Service Workbench on AWS

Implementation Guide

Service Workbench on AWS: Implementation Guide

Table of Contents

Service Workbench on AWS

AWS Solutions Implementation Guide

Publication date: *December 2020 (last update (p. 28): April 2021)*

This implementation guide describes architectural considerations and configuration steps for deploying Service Workbench on AWS 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.

The guide is intended for IT architects, developers, DevOps, and data analysts who have practical experience architecting in the AWS Cloud.

Overview

Service Workbench on AWS is a cloud solution that enables IT teams to provide secure, repeatable, and federated control of access to data, tooling, and compute power that researchers need. With Service Workbench, researchers no longer have to worry about navigating cloud infrastructure. They can focus on achieving research missions and completing essential work in minutes, not months, in configured research environments.

With Service Workbench, researchers can quickly and securely deploy research environments and conduct experiments with peers from other institutions. By automating the creation of baseline research setups, simplifying data access, and providing price transparency, researchers and IT departments save time, which they can reinvest in following cloud best practices and achieving research reproducibility.

See Service Workbench on AWS web page for more information.

Cost

You are responsible for the cost of the AWS services used while running this solution. At the date of publication, the cost for running this solution with the default settings in the US East (N. Virginia) Region is shown in the table below. Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Base cost of the Service Workbench on AWS platform:

AWS Service	Monthly Cost
Amazon API Gateway (100,000 requests/month)	$3.50
AWS Lambda (10,000 requests/month)	Free
Amazon DynamoDB (10MB)	Free
Amazon S3 (2 TB)	$47

Compute environments deployed through Service Workbench on AWS will incur additional costs based on your usage. An example using 2,000 compute hours per month is given in the table below.

AWS Service	Monthly Cost
Amazon EC2 (2,000 compute hours/month)	$384
Amazon SageMaker (2,000 compute hours per month)	$1,614
Amazon EMR (2,000 compute hours/month)	$480

Architecture overview

Service Workbench on AWS is a serverless environment that is deployed using an event-driven API framework. Its components are spread across Lambda instances, static webpages using Amazon

CloudFront and Amazon Simple Storage Service (Amazon S3). Service Workbench relies on AWS Service Catalog to host and manage AWS CloudFormation templates that defines the workspaces.

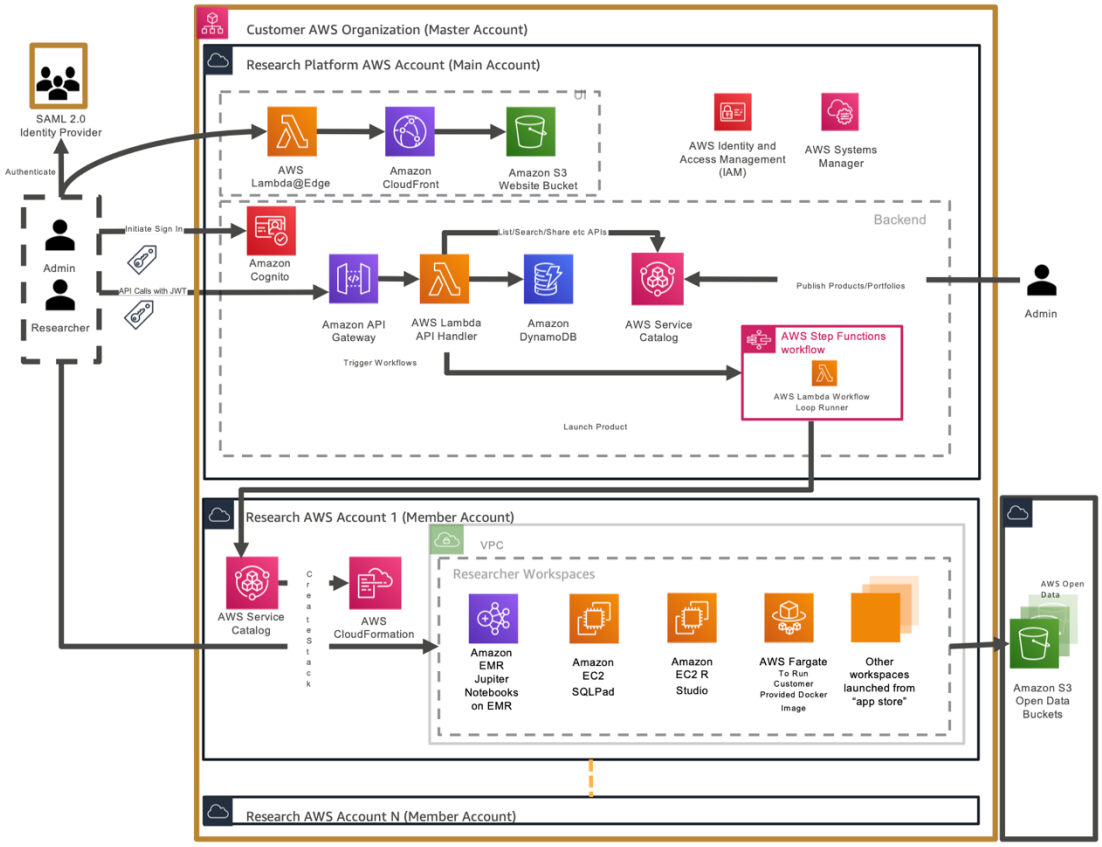

Figure 1: Service Workbench on AWS architecture

The Service Workbench on AWS administrator deploys the solution platform into the main AWS account, as indicated in Figure 1.

Researchers at the institution navigate to the Service Workbench on AWS web portal (provided by Amazon CloudFront backed by an Amazon S3 Static Website bucket) which serves as the entry point for launching, connecting to, and terminating compute workspaces defined by AWS CloudFormation templates. The web frontend authenticates using Amazon Cognito and leverages Amazon API Gateway to invoke the solution's microservices (AWS Lambda functions and AWS Step functions).

These microservices interact with Amazon DynamoDB to manage the content, users and AWS accounts in AWS Organizations to access data in S3 and instantiate out-of-the-box compute instances for Amazon EMR, Amazon SageMaker, Amazon EC2 with Windows and Linux operating systems. The CloudFormation templates for these compute instances are hosted in AWS Service Catalog for flexibility and to allow the simple addition of custom templates.

Solution components

User web portal

Service Workbench on AWS includes a web portal used by both Administrators and Research users. It is deployed into the main account using Amazon CloudFront and a static web site hosted in an Amazon S3 website bucket. The user web portal uses Amazon DynamoDB to store metadata about users, available datasets, and compute workspaces.

Research Studies S3 bucket

Service Workbench on AWS provides the ability to create and share Research Studies. Studies are hosted in an internal Amazon S3 bucket that is created during installation. The User Web Portal provides the ability to create and manage Studies, including the upload of data.

AWS Open Data integration

Service Workbench on AWS can access AWS Open Data via API calls. The User web portal allows researchers to select data sets to use in their compute environments.

CI/CD Pipeline

Service Workbench on AWS includes a CI/CD pipeline that is used to install the application in your account. This pipeline uses an AWS CodeBuild project that is created during deployment of the solution using the CloudFormation template.

Design considerations

Regional deployments

This solution uses the AWS CodeBuild, Amazon CloudFront, and Amazon Service Catalog services, which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where this service is available. For the most current availability by Region, refer to the AWS Service Region Table.

Additionally, this solution requires Amazon CloudFront support for log delivery to S3 buckets, which is not currently available in all regions. See Amazon CloudFront documentation, *Buckets for Logs* section, for a list of regions where this is not available.

AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of Service Workbench on AWS in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment:

service-workbench-on-aws.template: Use this template to launch the solution and all associated components. The default configuration deploys AWS Lambda, Amazon DynamoDB, and other services, but you can customize the template to meet your specific needs.

Automated deployment

Before you launch the solution, review the architecture, configuration, network security, and other considerations discussed in this guide. We recommend that you test the instance size requirements as well. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 45 minutes.

Prerequisites

AWS Organizations

If you wish to enable Service Workbench on AWS to create new AWS accounts via the Create AWS Account feature, then AWS Organizations must be enabled.

Cost Explorer

In order to see any actual cost in dashboards and workspaces, the main account must be set up in Cost Explorer. The main account holds the AWS Organization which creates member accounts.

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters.
- Review the other template parameters, and adjust if necessary.

Step 2. Post-launch tasks

- Tasks in the AWS Management Console
- Tasks in the Service Workbench on AWS user web portal

Step 1. Launch the stack

This automated AWS CloudFormation template deploys Service Workbench on AWS in the AWS Cloud.

> **Note**
> You are responsible for the cost of the AWS services used while running this solution. For more details, visit to the Cost (p. 2) section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the AWS CloudFormation template.

Alternatively, you can download the template as a starting point for your own implementation.

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 the Amazon Service Catalog service, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Amazon Service Catalog is available. For the most current availability by Region, refer to the AWS Service Region Table.

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 STS Limits 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
CreateMachineImages	`true`	Generate Amazon Machine Images (AMIs) during deployment. See Amazon Machine Images (AMI) for more information.
EnvironmentType	`dev`	Enables grouping multiple environments with a common name. Use dev, demo, qa, or prod.
ServicePortfolio	`true`	Create Service Catalog product and portfolio entries.
SolutionName	`swb`	Included in all resource names created by the deployment. Limited to 7 characters.
StageName	`test`	Allows multiple Service Workbench on AWS installations into the same account. Included in resource names created by the deployment. Should be limited to 5 characters.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next.**

8. 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.

9. Choose **Create 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 **CREATE_COMPLETE** status in approximately 20 minutes.

   Note
   Service Workbench on AWS creates four additional stacks during deployment: `Infrastructure`, `Backend`, `edgeLambda`, and `postDeployment`.

Step 2. Post-launch tasks

Gather information for post-launch tasks

Configuration of Service Workbench on AWS requires output values created during deployment. Use the steps below to gather this information.

1. Sign in to the **AWS Management Console**
2. Navigate to **CloudFormation**

3. Click on **View stacks**
4. Select the `infrastructure` stack created by Service Workbench on AWS
5. Click on **Outputs**
6. Note down the `WebsiteUrl` output.
7. Select the backend stack created by Service Workbench on AWS
8. Note down the `ApiHandlerRoleArn` and `WorkflowLoopRunnerRoleArn` outputs.
9. Navigate to **AWS Systems Manager**
10. Click on Parameter Store
11. Select the `/user/root/password` parameter created by Service Workbench on AWS.
12. Click **Show** to reveal the password and note it down.
13. Navigate to **Amazon EC2**.
14. Click on **AMIs** in the left navigation panel.
15. Note down the Names and IDs for all AMIs created by Service Workbench on AWS.

Onboard AWS account

Service Workbench on AWS enables users to launch compute resources into an AWS account. Onboarding the AWS account creates the necessary roles and permissions for Service Workbench on AWS to use.

1. Sign in to the **AMS Management Console**.
2. Navigate to **CloudFormation**.

3. Download onboard-account.cfn.yml by clicking this link and saving the file.
4. Click **Create stack**, and upload the onboard-account.cfn.yml file.

5. Click **Next**.
6. Under **Parameters**, review the parameters and modify them as necessary.

Parameter	Default	Description
Namespace	*<Requires input>*	An environment name that will be prefixed to resource names; use the stage name.
CentralAccountId	*<Requires input>*	The account number of the AWS account where the solution is deployed.
ExternalId	*<Requires input>*	A unique ID used to identify this account; use workbench.
VpcCidr	10.0.0.0/16	IP range (CIDR notation) for this VPC
VpcPublicSubnet1Cidr	10.0.0.0/19	IP range (CIDR notation) for the public subnet in the 1st Availability Zone
ApiHandlerArn	*<Requires input>*	The ARN of apiHandler role noted down earlier.
LaunchConstraintPolicyPrefix	*	Customer managed policy name prefix to use when creating a launch constraint role in the on-boarded account
LaunchConstraintRolePrefix	*	Role name prefix to use when creating a launch constraint role in the on-boarded account
WorkflowRoleArn	*<Requires input>*	The ARN of workflowRunner role

8. Choose **Next**.
9. On the Configure stack options page, choose **Next**.
10. 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.
11. Choose **Create stack** to deploy the stack.
12. 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 5 minutes.
13. When the stack reaction is complete, click on **Outputs**.
14. Note down the output values for `CrossAccountEnvMgmtRoleArn`, `CrossAccountExecutionRoleArn`, `EncryptionKeyArn`, `VPC`, and `VpcPublicSubnet1`.

Configure accounts, users, projects, and indexes

1. Open a web browser.
2. Open a web browser and navigate to `WebsiteUrl` as noted down earlier.
3. Sign in to **Service Workbench on AWS web portal**.

a. Username: `root`

b. Password: the `/user/root/password` as noted down earlier.

Note
Logging in as the root user is not recommended after initial step. The following steps include creating an administrator user for future use.

4. Add AWS Account to Service Workbench on AWS

   a. Click **Accounts** in the left navigation pane.

   b. Click **AWS Accounts**.

   c. Click **Add AWS Account**.

   d. Provide account details.

	Value	Notes
Account Name	*<enter text>*	
AWS Account ID	*<enter text>*	Provide the number of the AWS account that you deployed Service Workbench on AWS into.
Role Arn	*<enter text>*	Provide `CrossAccountExecutionRoleArn` as noted down earlier.
AWS Service Catalog Role Arn	*<enter text>*	Provide `CrossAccountEnvMgmtRoleArn` as noted down earlier.
External ID	*<enter text>*	Enter `workbench`.
Description	*<enter text>*	
VPC ID	*<enter text>*	Provide `VPC` as noted down earlier.
Subnet ID	*<enter text>*	Provide `VpcPublicSubnet1` as noted down earlier.
KMS Encryption Key ARN	*<enter text>*	Provide `EncryptionKeyArn` as noted down earlier.

5. After adding the account, click **Index** tab.

Projects and Indexes (cost centers) form a hierarchy under Accounts. Each Account can have multiple Indexes and each Index can have multiple Projects. Projects are attached to Users, so you must create the Projects first.

6. Click **Add Index.** See Figure 2.

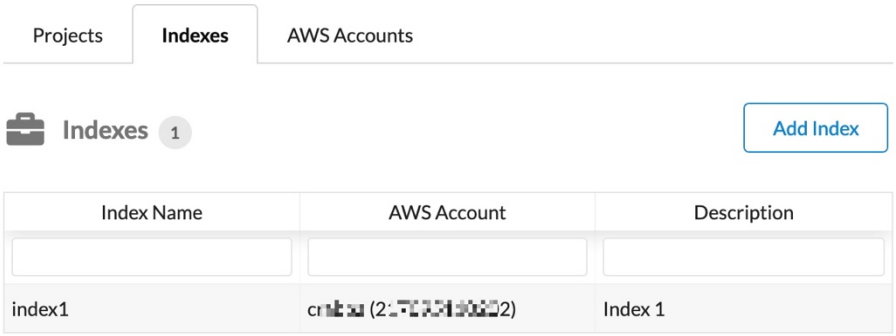

Figure 2: Create an Index

7. Provide an Index ID, AWS Account ID, and Description; then click **Add Index**.
8. Click **Projects** tab. See Figure 3.

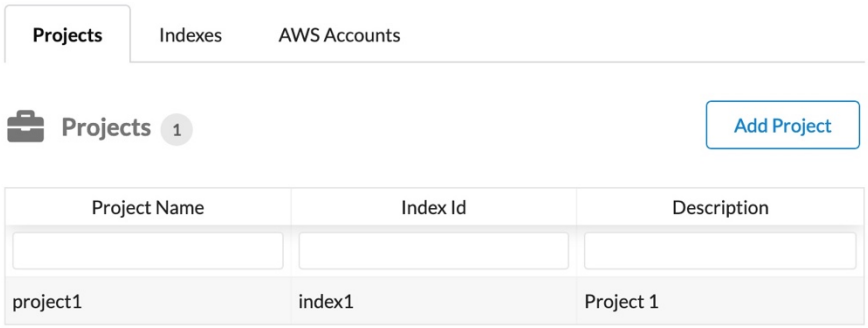

Figure 3: Create a Project

9. Click **Add Project**.
10. Provide a Project Name, select an Index ID, provide a Description, and select Project Admins; then click **Add Project**.
11. Click **Users** in the left navigation panel.
12. Click **Roles** tab. See Figure 4.

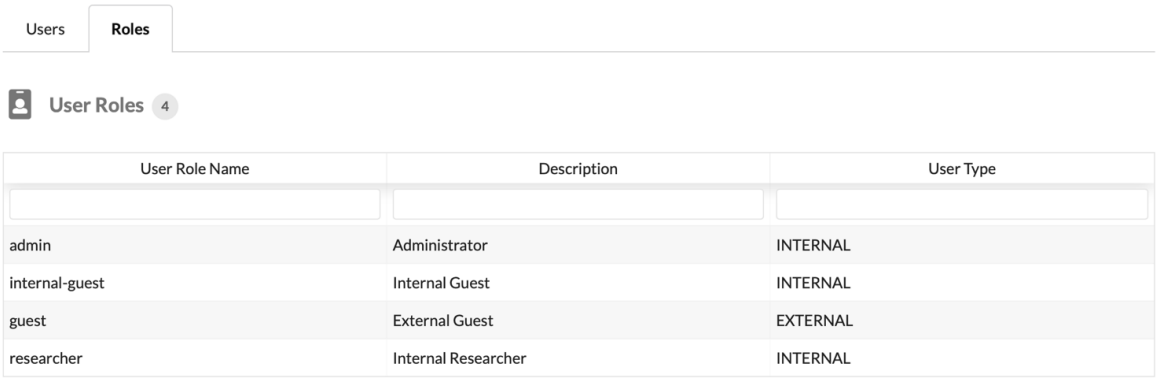

Figure 4: Roles tab

Roles are used to control access to Workspace Types in Service Workbench on AWS.

13. Click **Users** tab.
14. Click **Add Local User**

> **Note**
> Service Workbench on AWS supports local users for rapid setup for configuration, evaluation, and testing. For production usage, configure Federated users.

15. Create an administrator local user.
    a. For **UserRole**, select admin.
    a. For **Status**, select Active.
    b. Click **Add Local User**.

16. Click **Logout** in the top right corner of the page.
17. Log in as the administrative user created in the previous step.

Import and test Service Catalog Products

Service Workbench on AWS uses AWS Service Catalog to manage different types of computation resources available for researchers to use through the platform. Each product can have multiple size configurations defined.

When Service Workbench on AWS is deployed, an AWS Service Catalog portfolio is created with four commonly used products: Amazon SageMaker, Amazon EC2 for Windows, Amazon EC2 for Linux and Amazon EMR. These definitions must be imported into Service Workbench on AWS and configured before they can be deployed.

Access to Workspace Type configurations can be controlled based on the user's Role.

> **Note**
> Creating Workspace Types requires the IDs for the AMIs created during deployment. The AMI IDs are collected the "Tasks in AWS Management Console" section above.

1. Click **Workspace Types** in the left navigation bar.

AWS Service Catalog products available for import are listed. See Figure 5.

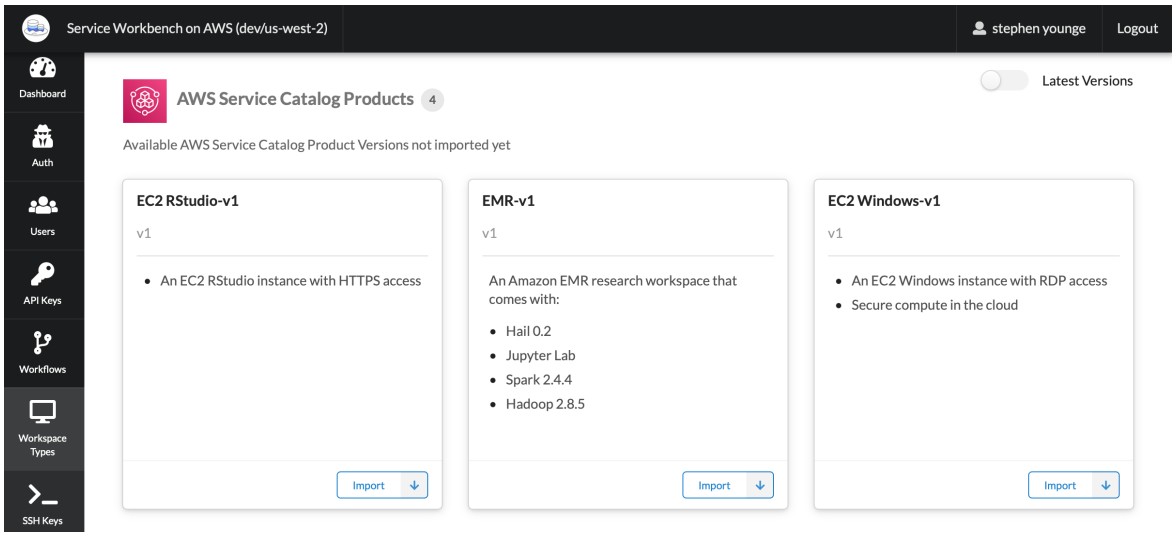

Figure 5: List of AWS Service Catalog products available for import

2. Click **Import** on the AWS Service Catalog product that you wish to import.
3. Provide Basic information.
4. Click **Import Workspace Type**.
5. Click **Add Configuration**.

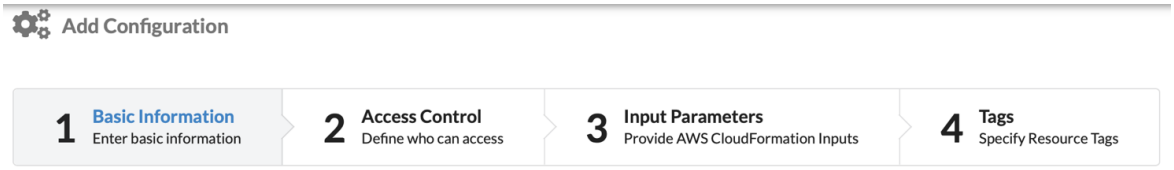

Figure 6: Steps for creating a Workspace Type Configurations

6. Input **Basic Information**
   a. Provide a unique ID for the configuration.
   b. Provide a Name for the configuration.
   c. Provide a Description for the configuration.
   d. Provide a Cost Estimate for the configuration. This field is optional.
   e. Click **Next**

7. Input **Access Control**
   a. Add `admin`, `researcher` roles to Roles Allowed.
   b. Roles Not Allowed is optional.
   c. Click **Next**.

8. Input **Parameters** to configure the environment.

The Input Parameters tab lists parameters common to all workspace types, plus some parameters specific to the chose instance type.

Explanations and suggested values for common and instance-specific input parameters are provided in the tables below.

Common input parameters

	Value	Notes
AccessFromCIDRBlock	`${cidr}`	Choose `cidr` from dropdown list.
EncryptionKeyArn	`${encryptionKeyArn}`	Choose `encryptionKeyArn` from dropdown list.
EnvironmentInstanceFiles	`${environmentInstanceFiles}`	Choose `environmentInstanceFiles` from dropdown list.
IamPolicyDocument	`${iamPolicyDocument}`	Choose `iamPolicyDocument` from dropdown list.

	Value	Notes
KeyName	`${adminKeyPairName}`	Choose `adminKeyPairName` from dropdown list.
Namespace	`${namespace}`	Choose `namespace` from dropdown list.
S3Mounts	`${s3Mounts}`	Choose `s3Mounts` from dropdown list.
Subnet	`${subnetId}`	Choose `subnetId` from dropdown list.
VPC	`${vpcId}`	Choose `vpcId` from dropdown list.

EC2 Linux input parameters

Tip
Information on EC2 instance types is available at Amazon EC2 Instance Types.

	Value	Notes
AmiId	*<enter text>*	The AMI ID (from Amazon Management Console for EC2) for the Linux machine image.
EncryptionKeyArn	`${encryptionKeyArn}`	Choose `encryptionKeyArn` from dropdown list.
InstanceType	*<enter text>*	An EC2 instance type, e.g. `t3.small`.

EC2 Windows input parameters

	Value	Notes
AmiId	*<enter text>*	The AMI ID (from Amazon Management Console for EC2) for the Windows machine image.
DownloadInterval	*<enter text>*	An interval in seconds to wait between two downloads in case of recurring downloads.
InstanceType	*<enter text>*	An EC2 instance type, e.g. `t3.small`.
KeyName	`${adminKeyPairName}`	Choose `adminKeyPairName` from dropdown list.
Namespace	`${namespace}`	Choose `namespace` from dropdown list.
RaidDataVolumeSize	*<enter number>*	Size of the instance EBS volume.

	Value	Notes
RecurringDownloads	*<enter text>*	true or false
StopRecurringDownloadsAfter	`-1`	Duration after which to stop downloads. Enter -1 to never stop recurring downloads.

SageMaker input parameters

	Value	Notes
AutoStopIdleTimeInMinutes	*<enter text>*	Number of idle minutes before auto stop of the instance. 0 disables auto stop.

Amazon EMR Parameters

> **Tip**
> For EMR cluster sizing guidelines, see Amazon EMR Cluster Configuration Guidelines and Best Practices.

	Value	Notes
DiskSizeGB	*<enter number>*	EBS Volume size (GB) for each node; provide a value ≥10.
CoreNodeCount	*<enter number>*	Number of core nodes to provision (1-80)
MasterInstanceType	*<enter text>*	EMR master node EC2 instance type, e.g. `m5.xlarge`
Market	*<enter text>*	Which market to purchase workers on - ON_DEMAND or SPOT.
KeyName	*<enter text>*	SSH key pair to use for EMR node login
WorkerBidPrice	*<enter number>*	Bid price for the worker spot nodes. This is only applicable when Market = SPOT. Specify 0 for Market = ON_DEMAND.
WorkerInstanceType	*<enter text>*	EMR node ec2 instance type, e.g. `ml.c4.xlarge`.
AmiId	*<enter text>*	The AMI ID (from Amazon Management Console for EC2) for the EMR machine image.

9. After providing input parameters, click **Next**

10Input **Tags**. This step is optional.

11. Click **Add**
12. The Workspace Configurations tab is displayed showing the new configuration. Additional configurations can be added later.
13. Click **Done**
14. The AWS Service Catalog Products page is displayed. The new workspace type will have the status Not Approved. See Figure 7.

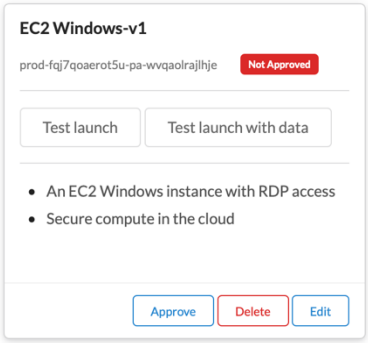

Figure 7: Pending workspace

15. Click **Test launch**
16. Provide a Name and Description, select a Project, and select the new Configuration.
17. Provide Restricted CIDR. Only IP addresses from within the specified CIDR block can access the Workspace. The default value corresponds to the computer's IP address.

    Note
    CIDR (Classless Inter-Domain Routing) is a method for allocating IP addresses and for IP routing. For more information, see Working with VPCs and subnets.

18. Click **Launch**
19. The Research Workspaces page will be displayed, with the new workspace in the Pending state. See Figure 8.

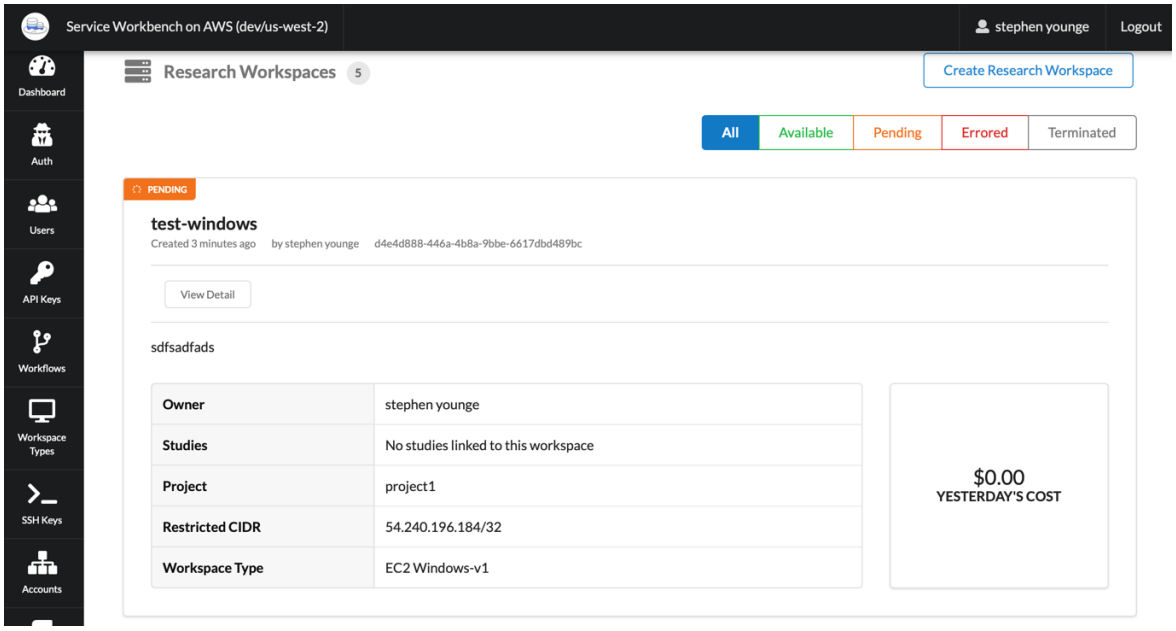

Figure 8: Research Workspaces page showing pending workspace

Note
Launching new workspaces can take five minutes or longer, depending on the resources.

20. When the Workspace status changes to Available, click **Connect**. The method used by Service Workbench on AWS to connect to the Workspace depends on the type of compute resources in the Workspace.

For **Linux** workspaces, Service Workbench on AWS uses SSH through Amazon EC2 Instance Connect. Copy and paste the suggested SSH command line into a terminal window.

For **Windows** workspaces, Service Workbench on AWS uses Remote Desktop Protocol (RDP).

For **Amazon SageMaker**, Service Workbench on AWS connects to the Jupyter notebook web interface.

For **Amazon EMR**, Service Workbench on AWS connects to the Jupyter notebook web interface. Log in with the default password, go-research-on-aws.

21. After connecting to the workspace correctly, click **Workspace Types** on the left navigation sidebar.

Note
If the workspace fails to launch, edit the configuration by returning to the Workspace Types page, editing the Workspace Type, then editing the Configuration.

22. If desired, create additional configurations for the workspace type by repeating the steps above.
23. Click **Approve** for the new workspace type. This makes the workspace type available to other users.
24. Repeat the steps in this section to import additional Service Catalog Products, if appropriate.

Create a research study

Service Workbench on AWS enables organizations to provide researchers with a centralized location to search for studies (data sets) and deploy research workspaces connected to them. Service Workbench on AWS supports three types of studies, described below.

- **My Studies:** Studies that are only available to the user that created them. A user can use this to work on datasets that are exclusive to them or that are used specifically for their research.
- **Organization Studies:** Studies that have been shared with the Organization. These could be data that had been collected by efforts of the organization or are licensed to the organization. It is possible to grant or deny users access to this data in order to comply with regulations or licensing restrictions on the data.
- **Open Data:** Publicly available studies published to Open Data on AWS.

To create a study in the Service Workbench on AWS web portal:

1. Click **Studies** in the left navigation pane.
2. Click **Create Study**.
3. Provide an ID for the Study.
4. Choose My Study or Organizational Study.
5. Enter a name for the Study in the Name field.
6. Enter a description for the Study in the Description field.
7. Select the Project that this Study relates to in the Project ID drop down field.
8. Click the **Create Study** button.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared 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 the AWS Security Center.

IAM Roles

AWS Identity and Access Management (IAM) roles enable 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. IAM roles created by this solution are named with the `SolutionName` and `Stage Name` that were specified during the initial deployment.

Amazon CloudFront

This solution deploys a web console 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, see Restricting Access to Amazon S3 Content by Using an Origin Access Identity in the *Amazon CloudFront Developer Guide*.

Amazon Virtual Private Cloud

The Amazon Virtual Private Clouds (VPCs) in this solution are designed to control and isolate network traffic into research compute environments. We recommend that you review the configuration and further restrict access as needed once the deployment is up and running.

Viewing Service Workbench logs in CloudWatch

Service Workbench has API Gateway access logging enabled. The logs are available in CloudWatch at the `/aws/api-gateway/lt;<name of your API>` log group:

Following is the format of the access logs:

```
{
"authorizer.principalId": "u-000000000000",
"error.message": "-",
"extendedRequestId": "ZuT4rGDNoAMFxXw=",
"httpMethod": "GET",
"identity.sourceIp": "22.22.222.22",
"integration.error": "-",
"integration.integrationStatus": "200",
"integration.latency": "79",
"integration.requestId": "67394741-90ae-4c6c-94fb-df8bf7be33ec",
"integration.status": "200",
```

```
"path": "/dev/api/user-roles",
"requestId": "468a1b4d-3015-4901-b749-37e4e0551029",
"responseLatency": "83",
"responseLength": "819",
"stage": "dev",
"status": "200"
}
```

Lambda logs are also available in CloudWatch with the default log group names `/aws/lambda/<lambda function name>`.

Available metrics

The default metrics for Lambda and API Gateway are available in CloudWatch. For the full list of available metrics, refer to:

- Working with AWS Lambda function metrics - AWS Lambda
- Amazon API Gateway dimensions and metrics - Amazon API Gateway Amazon API Gateway

Service Workbench does not emit any custom metrics.

Additional resources

AWS services

- Amazon API Gateway
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon EC2
- Amazon EMR
- Amazon S3
- Amazon SageMaker
- AWS CodeBuild
- AWS CloudFormation
- AWS Identity and Access Management
- AWS Lambda and Lambda@Edge
- AWS Service Catalog
- AWS Step Functions
- AWS Systems Manager
- Open Data on AWS
- AWS Key Management Service

Appendix A: Uninstall the solution

To uninstall Service Workbench on AWS, use the AWS Management Console or the AWS Command Line Interface to delete the stack.

Using the AWS Management Console

1. Sign in to the **AWS Management Console**.

2. Navigate to **Amazon CloudFormation**.
3. Delete the stack used to deploy the solution
4. Delete the four stacks created by the solutions installation:
   a. `Infrastructure`
   b. `Backend`
   c. `edgeLambda`
   d. `postDeployment`

5. Delete the stack created by the post-deployment account onboarding task.
6. Navigate to **Amazon S3**.
7. Manually empty and delete the S3 buckets created by the solution.
   a. `artifacts`
   b. `env-type-configs`
   c. `environments-bootstrap-scripts`
   d. `external-templates`
   e. `logging`
   f. `studydata`
   g. `website`

8. Navigate to **Amazon Service Catalog**.
9. Remove the Service Catalog portfolio.
   a. Choose the portfolio created by the solution
   b. Remove all products from the portfolio
   c. Remove all users from the portfolio
   d. Delete the portfolio

10 Remove the Service Catalog products created by Service Workbench on AWS.
   a. `EC2 Linux`
   b. `EC2 Windows`
   c. `EC2 RStudio`
   d. `EMR`
   e. `SageMaker Notebook`

11 Navigate to **Amazon Systems Manager**

12. Click **Parameter Store** in the left menu
13. Delete the entries created by Service Workbench on AWS:
    a. `/jwt/secret`
    b. `/user/root/password`

14. Navigate to **Amazon DynamoDB**
15. Click **Tables** in the left menu
16. Delete the tables created by Service Workbench on AWS.

Each table is prefixed with:

```
[stage-name]-[region-shortname]-[solution-name]
```

Accounts	DbPasswords	EnvironmentTypes
AuthenticationProviderConfigs	DbProjects	Indexes
AuthenticationProviderTypes	DbRevokedTokens	KeyPairs
AwsAccounts	DbStepTemplates	Locks
CostApiCaches	DbStudies	Passwords
DbAccounts	DbStudyPermissions	Projects
DbAuthenticationProviderConfigs	DbUserApiKeys	RevokedTokens
DbAuthenticationProviderTypes	DbUserRoles	StepTemplates
DbAwsAccounts	DbUsers	StorageGateway
DbCostApiCaches	DbWfAssignments	Studies
DbDeploymentStore	DbWorkflowDrafts	StudyPermissions
DbEnvironments	DbWorkflowInstances	UserApiKeys
DbEnvironmentsSc	DbWorkflows	UserRoles
DbEnvironmentTypes	DbWorkflowTemplateDrafts	Users
WorkflowTemplates	DbWorkflowTemplates	WfAssignments
DbIndexes	DeploymentStore	WorkflowDrafts
DbKeyPairs	Environments	WorkflowInstances
DbLocks	EnvironmentsSc	Workflows
		WorkflowTemplateDrafts

Source code

Visit the GitHub repository for the solution to download templates and scripts, and to share your customizations with others.

Contributors

The following individuals contributed to this document:

- Karl Camp
- Maggie Krummel
- Neel Sethia
- Robert Smayda
- Simon Woldemichael
- Stephen Younge
- Yanyu Zheng
- Yogesh Sharma

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.

Revisions

Date	Change
December 2020	Initial release
January 2021	Release version 1.4.4 - For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2021	Documentation updates