Service Workbench Post Deployment Guide

Table of Contents

Account types

Service Workbench uses *three* types of accounts. You will see these account names throughout the documentation.

- **Organizational**: Holds the AWS Organization which creates hosting accounts. Note that you may already have a method of obtaining AWS accounts supported by your organization. If this is the case, you will not create an organizational account or use the **Create Account** functionality within Service Workbench when onboarding a hosting account.
- **Main**: The account to which Service Workbench is deployed, serving as the central point of login and access to researcher workspaces.
- **Hosting**: Accounts that are established associated to the Service Workbench main account through the onboarding process to host the compute resources (Amazon SageMaker notebook instances, Amazon EC2 Windows and Linux instances, Amazon EMR clusters) associated to Service Workbench workspaces.
- **Devops**: This account can be used to store Appstream images and product AMIs centrally and shared across multiple SWB installations.

Read the following files in the source code documentation to learn more about the different types of AWS accounts within Service Workbench:

- `README.md`
- `main/solution/prepare-master-acc/README.md`

Enable Local Users

Local users are created only within the solution. Their credentials are stored in Amazon DynamoDB. This is the easiest way to install. The alternative is to integrate with an Active Directory.

Preparing the organizational account

Note: This section is required only if Service Workbench is to be used to vend accounts in the AWS Organization by using Create Account. If Service Workbench uses only billing accounts onboarded using Add Account, this section can be omitted.

In this step, you deploy the **prepare_master_acc** Separately Deployable Component (SDC) in the main/solution/prepare-master-acc directory. This creates a role in the organizational account that allows **AssumeRole**, and has the **main** account as its trusted entity. If you have deployed

Service Workbench in an **organizational** account, the **main** account is also the **organizational** account, and the trusted entity is the same account ID as the **organizational** account.

The default settings in this step are:

- **Main account ID**: The current AWS account ID
- **External ID**: The string **workbench**

These default values are sufficient if you are deploying Service Workbench to an **organizational** account, and the default profile has permissions for the **organizational** account, since the **Main** account is also the **organizational** account. If deploying Service Workbench to the **main** AWS account that is not the same account as the **organizational** account, you must create a configuration file to specify the **main** account ID and the profile to use. This profile must have permissions for the **organizational** account.

Creating a configuration file

If deploying Service Workbench in an account other than that accessed by the current default profile, create a stage-named configuration file in the main/solution/prepare-master-acc/config/settings directory by copying example.yml to <stage>.yml. Edit the file as appropriate:

- **awsProfile**: The AWS credentials profile with permissions for the organizational account.
- **mainAccountID**: The 12-digit AWS account ID for the main AWS account, where the solution is deployed.
- **externalId**: As desired. The string **workbench** is often used. This string is required when creating an AWS account within Service Workbench.

Enabling AWS Organizations in the organizational account

In the AWS Management Console, navigate to **AWS Organizations** to ensure that an organization exists for the **organizational** account. If it does not, then create a new one. There is no configuration setting. Service Workbench creates a new account in the AWS Organizations for this deployment, named after the **stage name** used at deployment.

Deploying the prepare organizational account SDC

To deploy the prepare_master_acc SDC:

- Read the main/solution/prepare-master-acc/README.md file.
- To deploy the **organizational** account SDC from the main/solution/prepare-master-acc directory, enter the following command:

```
pnpx sls deploy --stage <stage>
```

- To display the ARN of the **organizational role** from the same directory, enter the following command:

```
pnpx sls info --verbose --stage <stage>
```

The **organizational role ARN** is needed when adding accounts within Service Workbench.

***Note**: Running the convenience script scripts/master-account-deploy.sh <stage> performs the same steps as pnpx sls deploy.*

Confirming the creation of organizational role

The newly-created role contains **MasterRole**. It has two policies and trusts the main account.

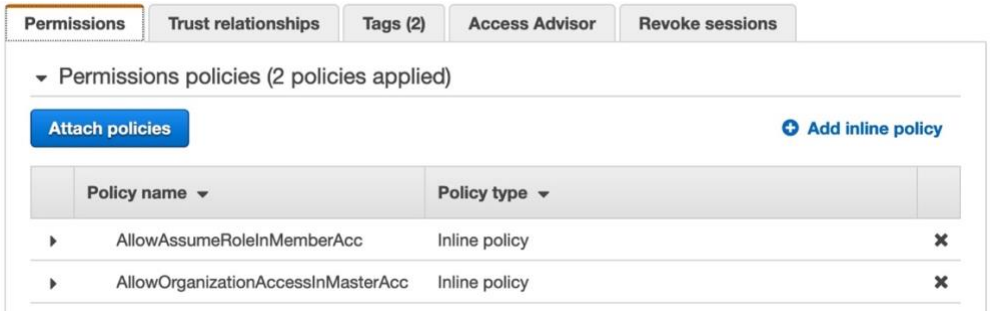


Figure 1: Permissions Policies

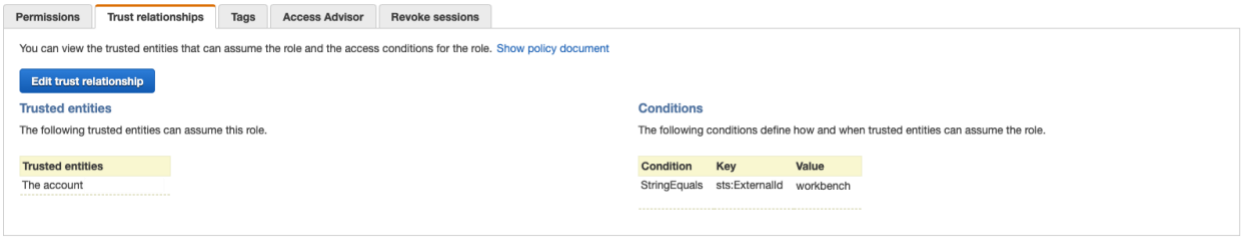


Figure 2: Edit Trust relationships

Prepare your Account for AppStream

Note: *Read this section only if you want to enable AppStream.*

AWS AppStream service limits may block resource creation in a new AWS account, or an account that has not yet hosted AppStream resources. The following actions will best prepare the hosting account.

1. **Required**: Go to AppStream 2.0 services (AWS Console), and choose **Get Started**. This will take you to a screen asking if you want to try out some templates. At this screen choose **Next**. This will only need to be done once for the account (it activates a role required for AppStream to be initiated).
2. **Recommended**: Launch and terminate at least one EC2 instance (size of instance and duration of run prior to termination do not matter), as this establishes the base compute service limits.
3. **Optional**: Open a support ticket within the hosting account to AWS AppStream service, noting your intention to create a fleet. This should only be necessary if you are creating a large fleet, but if the account is very new or has never had resources created prior to being onboarded as a hosting account, this is recommended.

Once prerequisites are complete and the CloudFormation stack has been created, go to AppStream on the AWS console of the hosting account. Go to **Fleet** and then choose the newly created fleet. Choose **Action**>**Start** to start the fleet. This step must be completed for any researcher workspace to be successfully launched within the hosting account.

After proceeding from the **Onboard AWS Account** Screen, you will be returned to the **Account listing** page with a status.

- **Up to Date** : The account has successfully onboarded and the version of the main and hosting account code is in sync.
- **Needs Update** : This status means that the main account code is more up to date than the hosting account code. There is an update button available and it is advised that you use this to bring the hosting account to the latest version.
- **Needs Onboarding** : The account has been onboarded via the Service Workbench UI but the CloudFormation template has not been generated
- **Errored** : The onboarding of the hosting account failed. Check the CloudFormation stack in the hosting account.
- **Pending** : The account onboarding process is ongoing. If a hosting account remains in pending status beyond 30 minute, it is likely that there is an issue onboarding the account. Check the CloudFormation stack in the hosting account.

Setting up an AppStream Image

Overview

This section describes the procedure of setting up an AppStream image with the following applications installed: Firefox, Notepad, PuttyGen, and Putty. AppStream should be built in the **main account**.

Pre-requisites

1. Navigate to AWS AppStream in your main account using the AWS Management Console.

2. Choose **Get Started** and then choose **Next**. This activates AppStream in your main account, following which you could proceed with the commands in `SETUP.md` file.

The image build process relies on the presence of the default VPC and outbound internet connectivity. If this is not available (typical of AWS Control Tower environments), then create a new default VPC.

Launching an AppStream Image Builder instance

1. Navigate to the `scripts/app-stream` directory.
2. Enter the following commands:

```
npm install
npm run start-image-builder -- <AWS Profile> <region> <Base image name>
<instance size>
# Example: npm run start-image-builder -- default us-east-1 AppStream-
WinServer2019-06-01-2021 stream.standard.medium
# If preferred you can choose the default base image name and instance
size by running this command: npm run start-image-builder -- default
us-east-1 default default
```

Note: Set up your AWS profile beforehand so that you have permission to launch an AppStream Image Builder instance in your AWS Account. If **enableAMISharing** is true, make sure you use devopsProfile for **<AWS Profile>** so the image builder will be created in the devops account.

3. Once the Image Builder is launched and ready, choose the URL provided in the terminal console. This opens the AppStream page on the AWS Console.

Building AppStream image

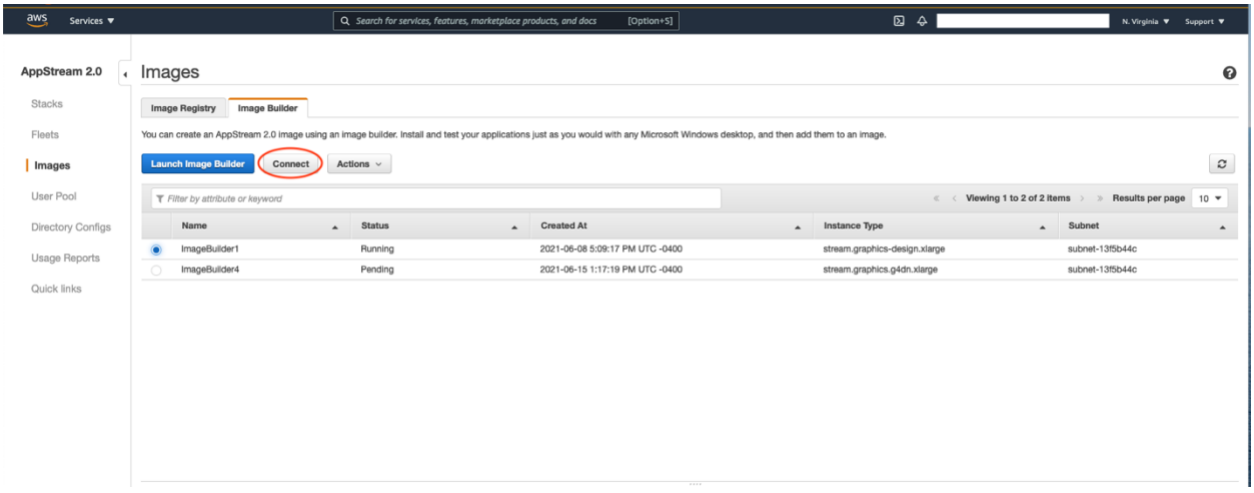


Figure: Create an AppStream image

1. On the AppStream page in the console, choose the AppStream image that was built in the previous step and choose **Connect**.
2. This opens a new tab in your browser. When prompted, log in as **Administrator**. This opens the Windows Desktop that you can interact with to create your AppStream image.
3. On the Windows Desktop, choose the `Start` button and enter `Windows Powershell`.
4. Right-click the application and choose `Run as administrator`.
5. Enter the following commands:

```
cd ~\Documents

# Pull the Image Builder script from Github
Invoke-WebRequest -Uri
https://raw.githubusercontent.com/awslabs/service-workbench-on-
aws/feat-secure-workspace-egress/scripts/app-stream/buildImage.ps1 -
OutFile buildImage.ps1

# Execute Image builder script
.\buildImage.ps1
```

6. At this point, the Image builder builds your image and the `Failed to reserve a session` message is displayed.
7. Log in to AppStream on the console again and wait till the AppStream image is built.

Create or add compute hosting accounts

After logging in as the **root** user for the first time, go to the **Accounts** page.

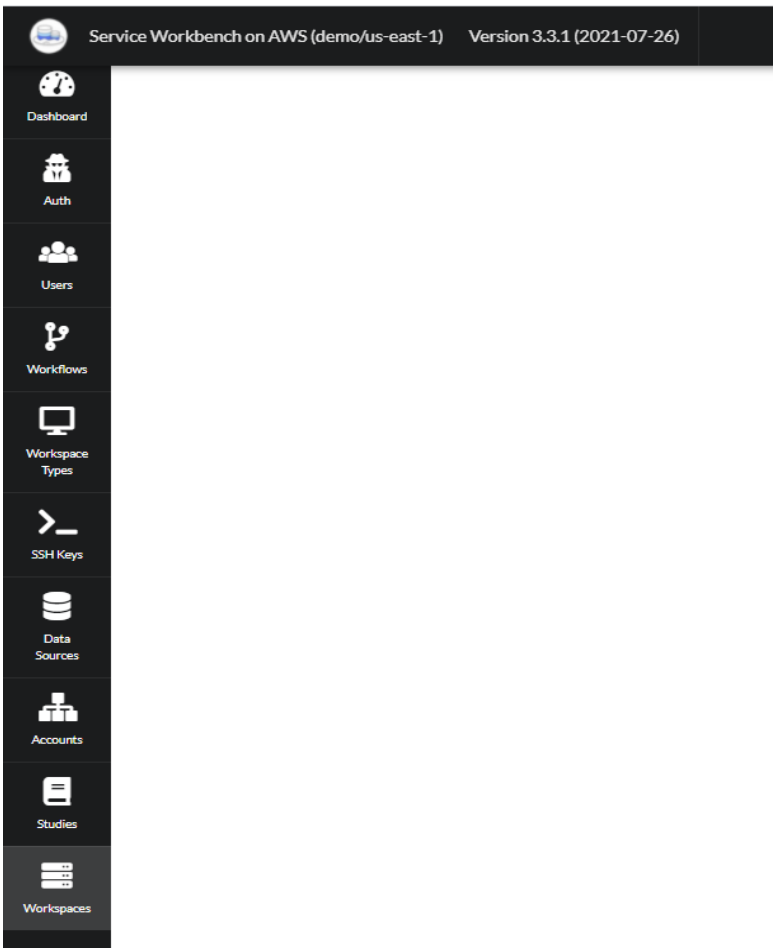


Figure: Service Workbench navigation bar

Service Workbench uses AWS accounts on this page for launching research workspaces. You can add existing AWS accounts or create new ones on the **Accounts** tab.

- **Create AWS Account**: Creates a new AWS account using AWS Organizations. Note that preparing organizational account steps outlined above are a pre-requisite should you require this capability.
- **Add AWS Account**: Associates an existing AWS account for purposes of hosting compute resources. This account can be responsible for its own billing.

Every user is linked to an account through a project and an index, so at least one account must be created or added before associating the first user to a project.

Note: If you do not need to create new AWS accounts from within Service Workbench, then skip to Add AWS Account section.

Create AWS Account

Prerequisites

Before creating an AWS account from Service Workbench, follow these guidelines:

- Configure an existing AWS account to be the organizational account for Service Workbench (steps outlined in *Preparing the organizational account* section). When Service Workbench creates new AWS accounts, billing for those accounts is applicable to the organizational account.

Creating a new Account

This creates a new **hosting** AWS account in the organization, whose billing goes to the **organizational** account.

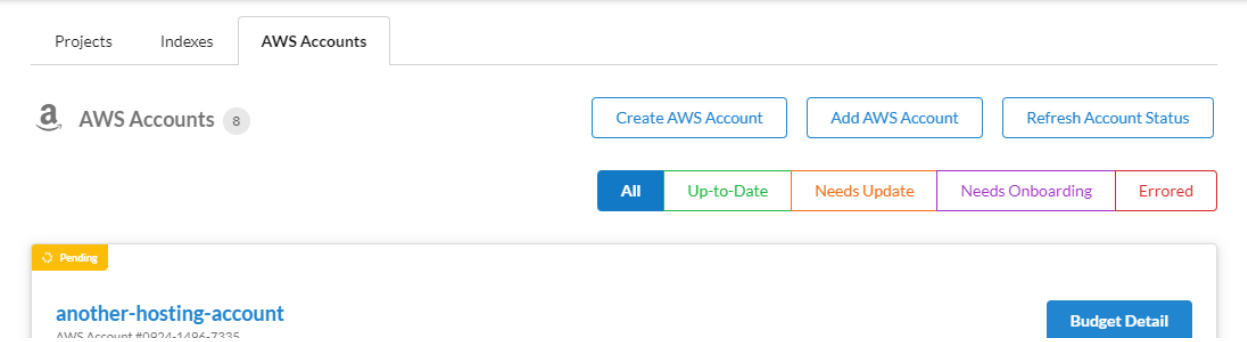


Figure: Create a new hosting account

To create an account:

1. In the Service Workbench console, choose **AWS Accounts** and then choose **Create AWS Account**.
    - In **Role ARN**, enter the **Master Role ARN** copied from the preparing the organizational account steps.
    - The email address that you specify here must be unique within the organization.
    - The **External ID** is **workbench** by default. See IAM for information on how to configure this to another value.
2. During processing, the following information displays in the **AWS Accounts** tab:
    - *'Trying to create accountID: xxx'*
    - A workflow in progress in **Workflows → Provision Account** (see Workflows)

    ***Note**: If instead you see an error message such as, 'Stop Internal State Account ID not found', check that there is an AWS Organization in the console of*

*your **organizational** account, if deploying Service Workbench in the **organizational** account. If you are deploying in a **Member** account, check and ensure that you followed the steps described in Prepare Organizational Account.*

- Optionally, in the AWS console, you can inspect the following resources deployed by this script:
  - In AWS CloudFormation, a stack **prep-master** is running. It creates the **Master** role and its output is the **Master Role ARN**.
  - In the AWS Organization, in the **Organizational** account (see IAM), the new account displays.
  - In IAM, the new **Master** role is created.

3. Once the account is created it is listed in **AWS Accounts**.

Add AWS Account

Adding an existing AWS account enables Service Workbench to launch research workspaces. The existing account alignment (standalone or associated to an an organization) determines the billing responsibility.

- On the **Accounts Page**, choose **AWS Accounts,** and then choose **Add AWS Account**.

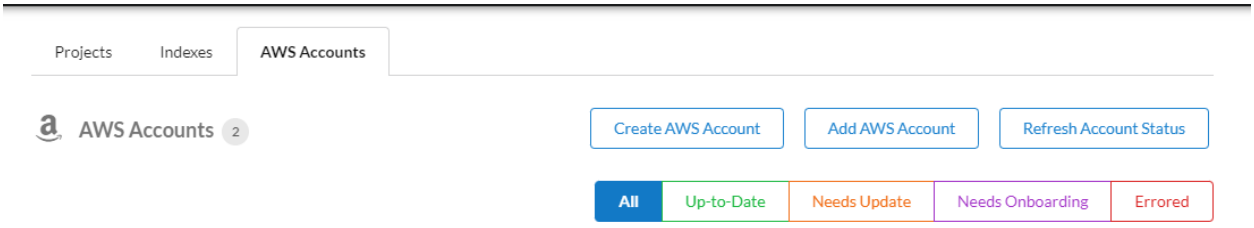


Figure: Add an existing account

- Create and run the AWS CloudFormation template:

If AppStream is not enabled

- Enter the **Account Name**, **Account ID** (12 digit AWS account ID for the account you wish to add), and **Description**.

Figure: Specify account details

- The **Onboard AWS Account** screen is displayed, where you can select either of the following options:
  - **I have admin access** : You have administrator-level access to the hosting account that is being onboarded. Selecting this and then proceeding launches the CloudFormation template within the hosting account. Note that you need to be logged into the AWS account console for the hosting account when selecting this option and proceeding.
  - **I do not have admin access** : The CloudFormation template is generated and you can then share the template to be run by the party that does have administrator access in the AWS account that you are onboarding.

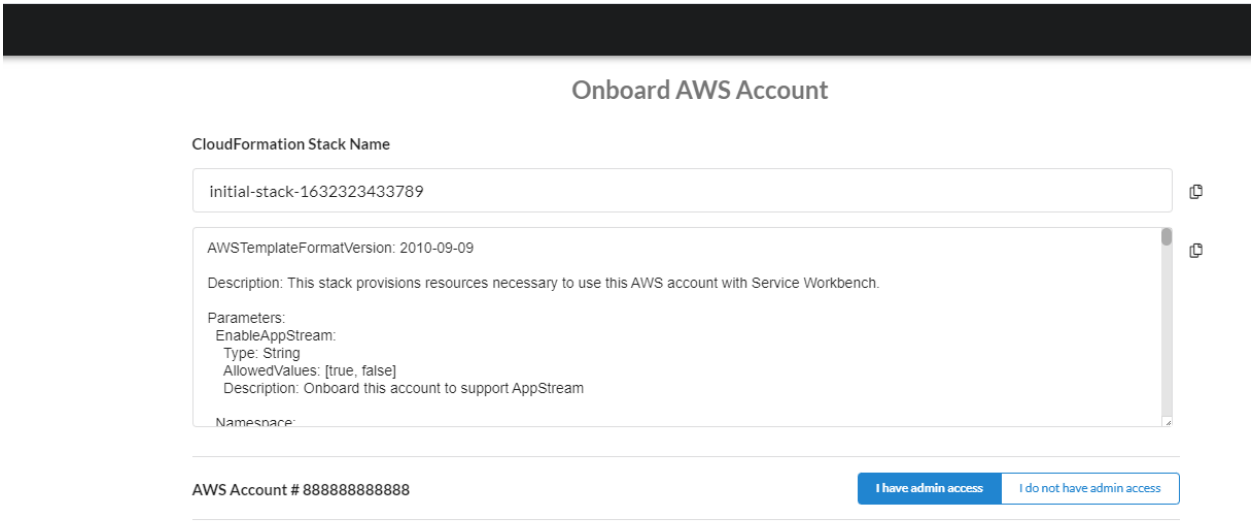


Figure: Onboard AWS account

If AppStream is enabled

If you have chosen to enable AppStream for your installation, there are additional values required when onboarding a hosting account.

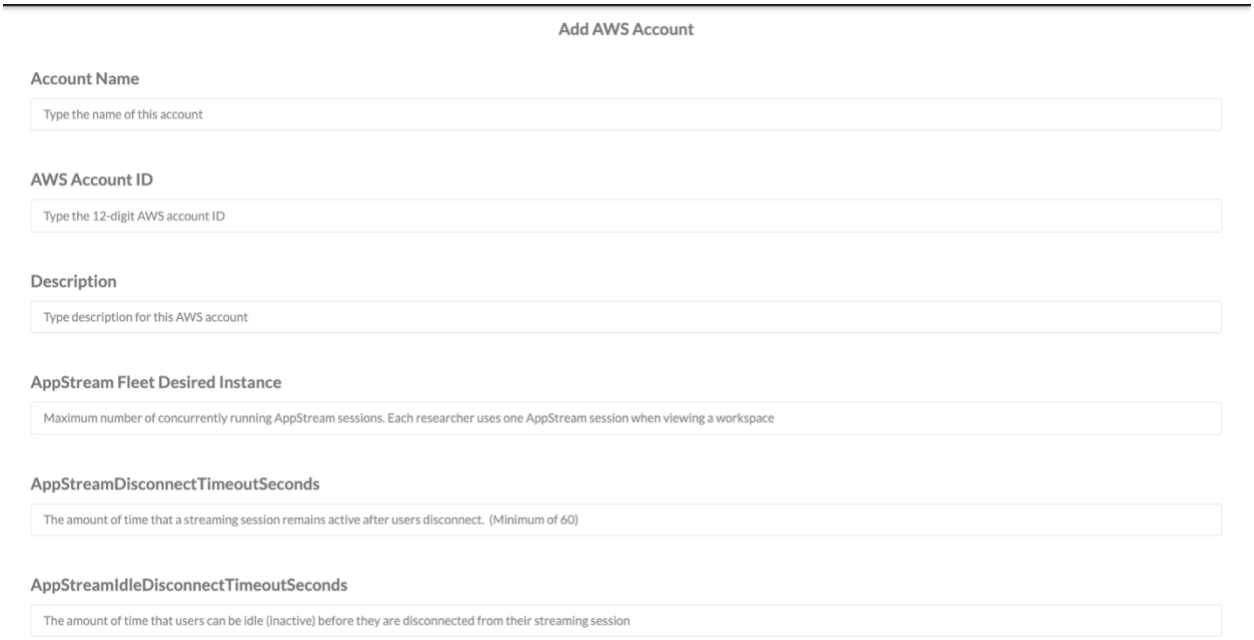


Figure: Add account when AppStream is enabled

- **AppStream Fleet Desired Instance**: The maximum number of concurrently running AppStream sessions allowed. If you set this to 5, that would mean that five workspaces can be viewed concurrently.
- **AppStreamDisconnectTimeoutSeconds**: With a minimum of 60 seconds, this is the amount between a researcher disconnection from a session (Manual Stop, Auto Stop, or Terminate) and the release of the AppStream instance that is supporting that session.
- **AppStreamIdleDisconnectTimeoutSeconds**: The amount of time that an AppStream session idle time (meaning no activity within the session) before the AppStream instance disconnects.

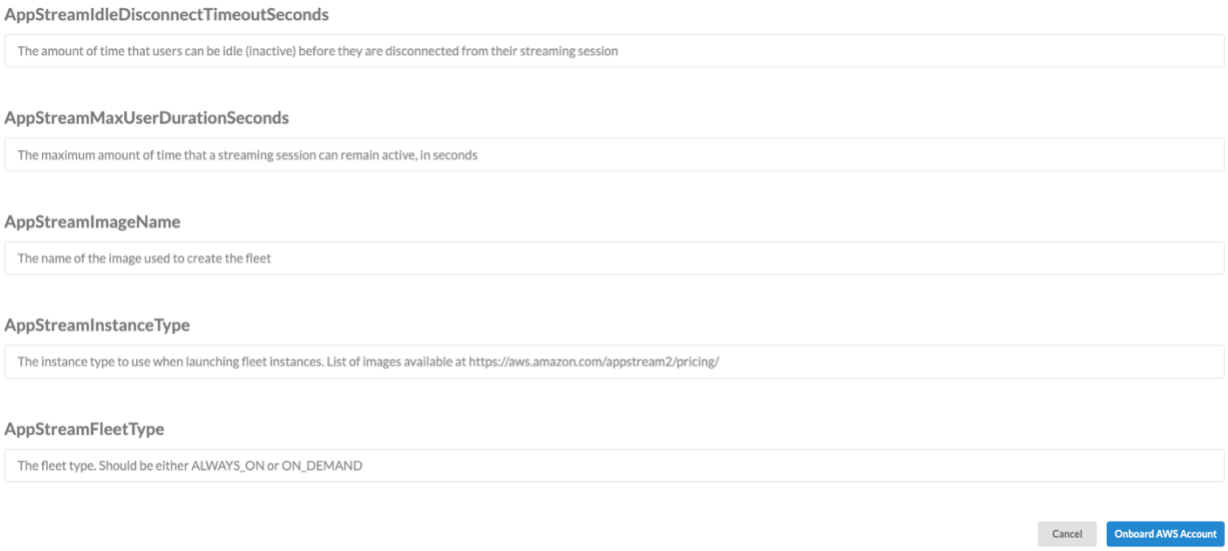


Figure: Add account when AppStream is enabled (contd..)

- **AppStreamMaxUserDurationSeconds**: The maximum amount of time for an AppStream session.
- **AppStreamImageName**: The exact image name produced when you follow the instructions to build your AppStream image. If **enableAMISharing** is true, make sure you use the AppStreamImageName from the image created in the devops account.
- **AppStreamInstanceType**: The instance type for your AppStream fleet. Note that these instance types are unique to AppStream. A complete list and specifications for valid instance types is available at https://aws.amazon.com/appstream2/pricing/
- **AppStreamFleetType**: `ALWAYS_ON` ensures that the desired number of instances remain available at all times. `ON_DEMAND` only runs the instances when required. It is a cost versus convenience choice.

  Note: If you needed to change these values later, you can do so through the AWS Console of the hosting account without negative impact to Service Workbench.

Once these options are specified, choosing **Onboard Account** displays the **Onboard AWS Account** screen. The same choice of Admin vs. No Admin access applies, but there are several important pre-requisites to complete before proceeding.

Creating RStudio ALB workspace

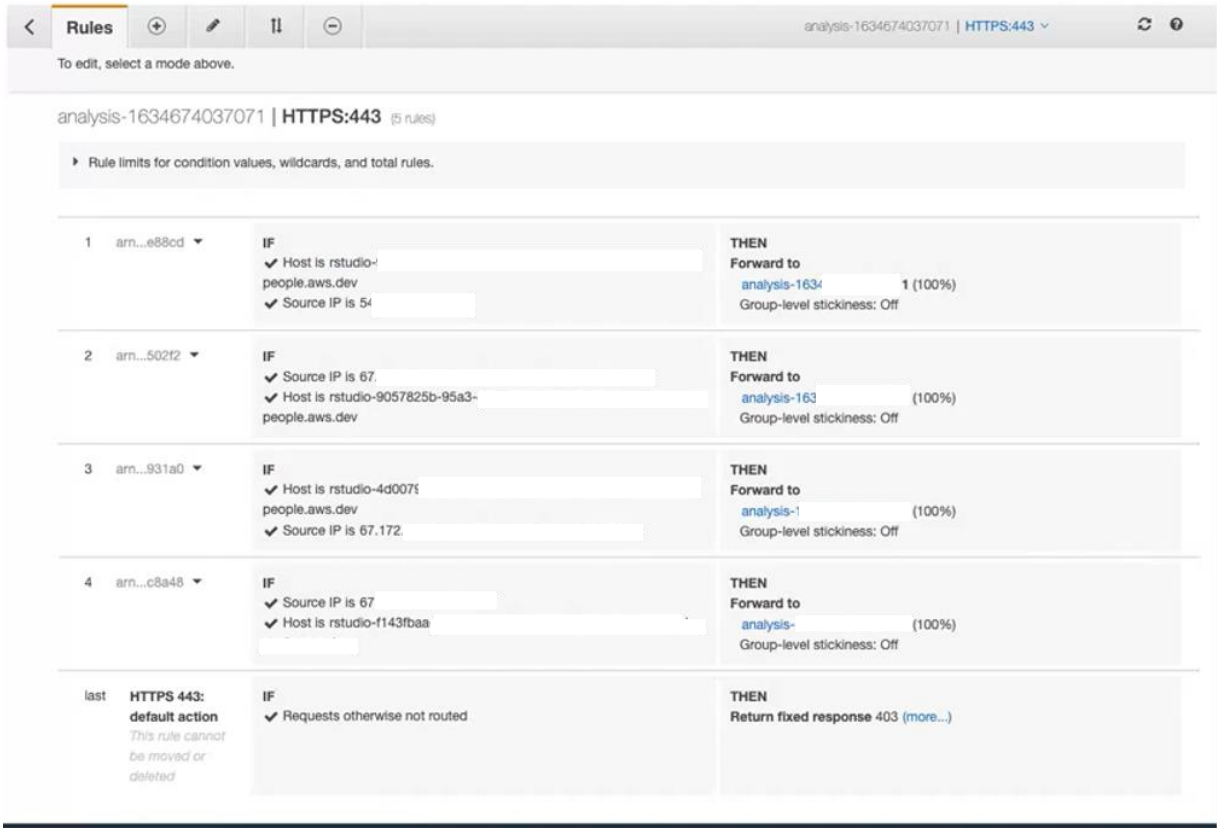
Accessing RStudio workspace

You can access the RStudio workspace type by using the template and AMI provided in AWS partner's repository. RStudio legacy and RStudio ALB both use custom domain names. You can define custom domain name in `/main/config/settings/<stage.yml>`.

Important: The legacy RStudio workspace type will be deprecated soon in a future release.

Application load balancing for RStudio ALB workspace

When you create RStudio ALB workspace, an ALB is created that can host upto 100 workspaces per hosting account. Per workspace, you can have upto 4 CIDR blocks (IP ranges) for port 443 only. This is updated in the hosting accounts. If there are no ALBs for that hosting account and you have created the very first workspace, then ALB will be created for the first time and it will be common for every workspace that you create subsequently. Once you terminate all the workspaces and you terminate the last workspace for that hosting account, the ALB is deleted. An ALB is created for minimum one active workspace.



Each workspace type corresponds to one listener rule and there can be 100 such rules. In each rule, users can specify the IP ranges that user wants to allowlist. You can specify only upto 4 IP

ranges per listener rule. For more information about ALB, refer to Application Load Balancer (ALB).

Limitations

1. With RStudio ALB implementation, you can create only upto 100 workspaces per hosting account.
2. Per workspace, you can specify only upto 4 CIDR blocks for port 443.

Create Indexes and Projects

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

After you create an account in the **Accounts** tab of the administrative interface, create an **Index** that links to the Account, by selecting the **Account ID** from the drop-down list.

1. On the **Indexes** tab, click **Add Index**. See the figure below.

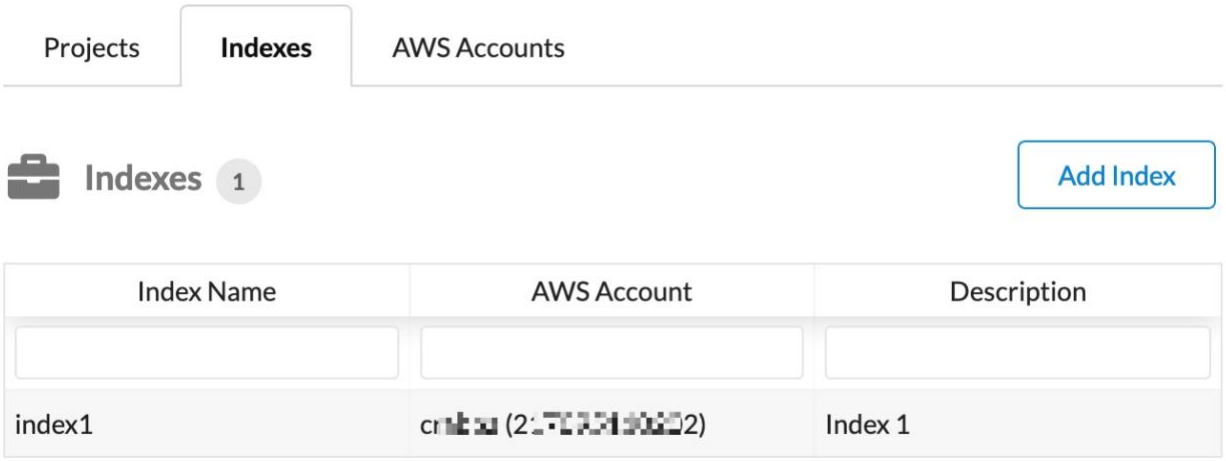


Figure : Create an Index

2. Create a **Project** that links to the new Index. In the **Projects**' tab, choose **Add Project**.

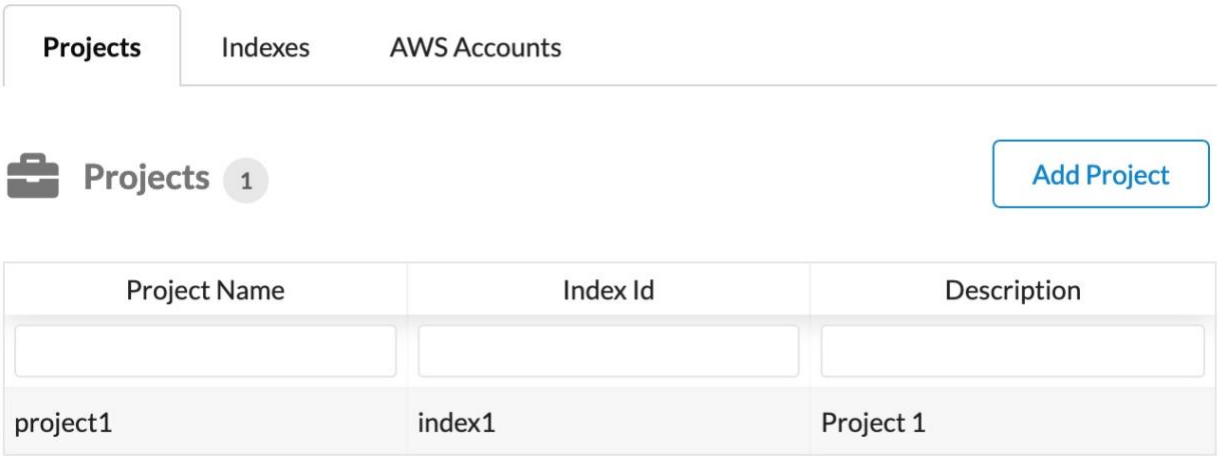


Figure: Create a Project

Create an Administrator User

Once you create an account and an index and project, you must create an administrator user in the **Users** tab.

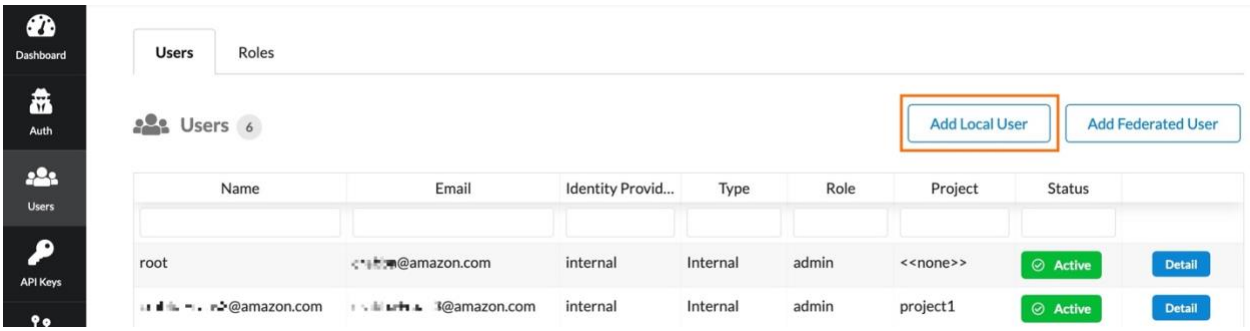


Figure: Create an Administrator

***Note**: A root user account will already be created, however, you must not routinely use the root user account.*

For testing purposes, you can create a local user by clicking '**Add Local User**'. Assign the user the administrator's role, and associate the user with the **Project** you created, and set the status to '**Active**'. See **Figure 8**.

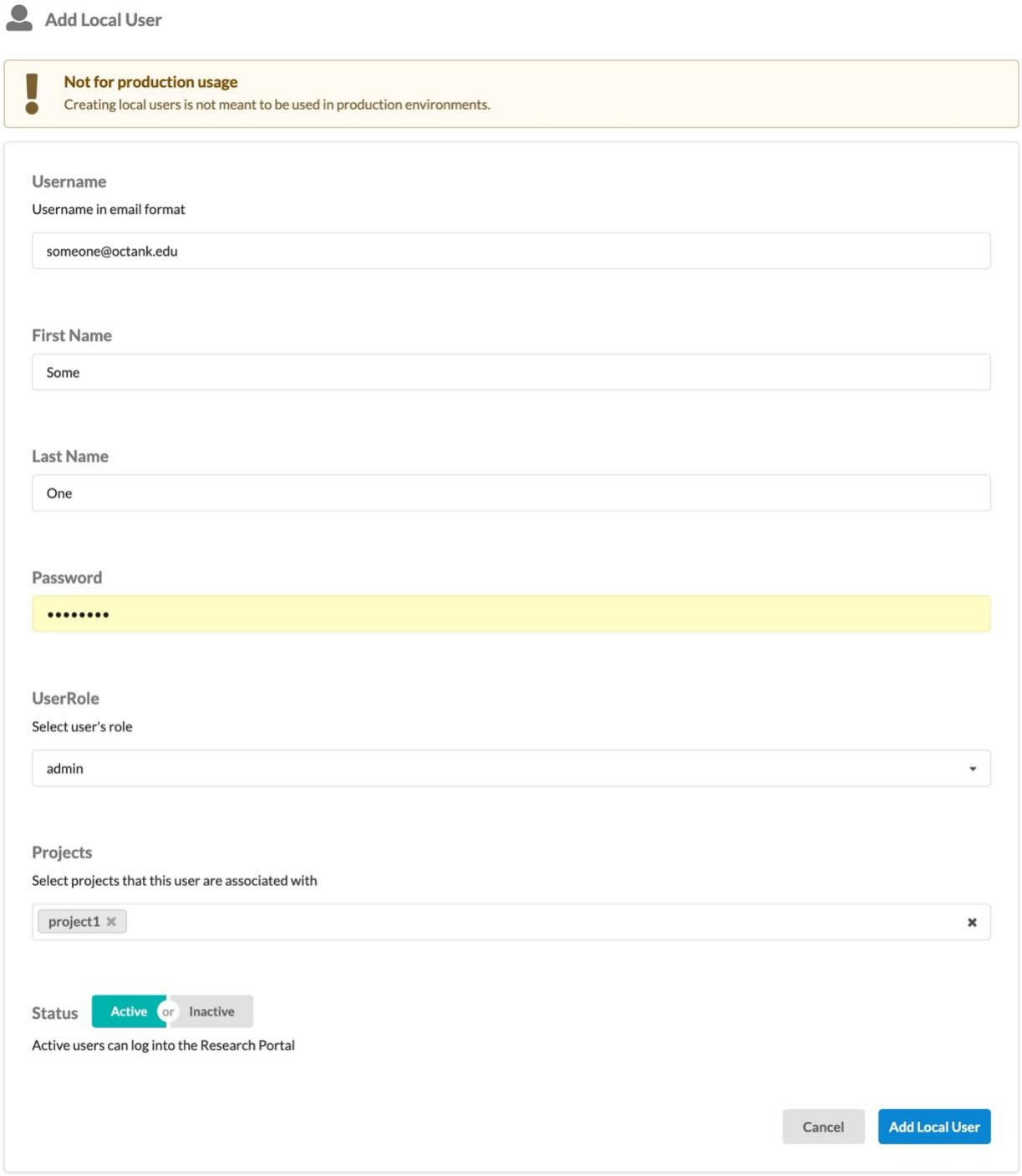


Figure: Add Local User

In prod environments we highly recommend using an IDP. For more details, refer to the ***Service Workbench Configuration Guide*.**

Import Service Catalog Products

Service Workbench uses AWS Service Catalog to manage different types of computation resources available for researchers to use through the platform.

With AWS Service Catalog integration, Service Workbench allows Admin users to create and manage catalogs of IT services that are approved for use on AWS. These IT services can include everything from virtual machine images, servers, software, and databases to complete multi-tier application architectures.

With this integration, Service Workbench helps organization to centrally manage commonly deployed IT services, and helps achieve consistent governance and meet compliance requirements, while enabling users to quickly deploy only the approved IT services they need.

When Service Workbench is deployed, an AWS Service Catalog portfolio is created by default with four commonly used products: Amazon SageMaker, Amazon EC2 for Windows, Amazon EC2 for Linux and Amazon EMR. The **administrator** needs to import and configure these products using Service Workbench user interface before they can be deployed. If you want to include additional custom products in the AWS Service Catalog portfolio, complete these steps:

1. Add the AWS CloudFormation template in the following directory:

   ```
   addons/addon-base-raas/packages/base-raas-cfn-
   templates/src/templates/service-catalog
   ```

2. Add the AWS CloudFormation template file name in the productsToCreate list in the following location:

   ```
   addons/addon-base-raas/packages/base-raas-post-
   deployment/lib/steps/create-service-catalog-portfolio.js
   ```

3. Deploy Service Workbench.

Import a Product

In this step, you import a pre-defined product, configure parameters to be used for product launch, and approve the configured product to be used. The following sections use Amazon EC2 Linux as an example, followed by setting different configuration required for Amazon EC2 Windows, Amazon SageMaker and Amazon EMR.

Prerequisites

Ensure the following prerequisites are met in order to import a product.

Creating AMI

Make sure you completed the step, deploy the Machine Images SDC as part of the deployment process.

To check if AMIs were created successfully:

1. Navigate to Amazon EC2.
2. Select the **AMI** tab.
3. Note down the four AMIs created for Amazon EC2 Linux, Amazon EC2 Windows, Amazon EMR, and Amazon EC2 RStudio.
4. Copy the AMI IDs and use for workspace import and configuration. Alternatively, you can also copy these AMI IDs from the terminal when the machine-images SDC is deployed.

***Note**: If you run the machine images SDC multiple times, duplicated AMIs are created. This is okay and will not affect any Service Workbench functionalities. You can choose to remove the duplicates to avoid confusion or leave them.*

Viewing AWS Service Catalog portfolio

1. Log in to Service Workbench UI as an **administrator**.
2. Navigate to **Workspace Types** tab. Four AWS Service Catalog Products display as shown in **Figure 9**.

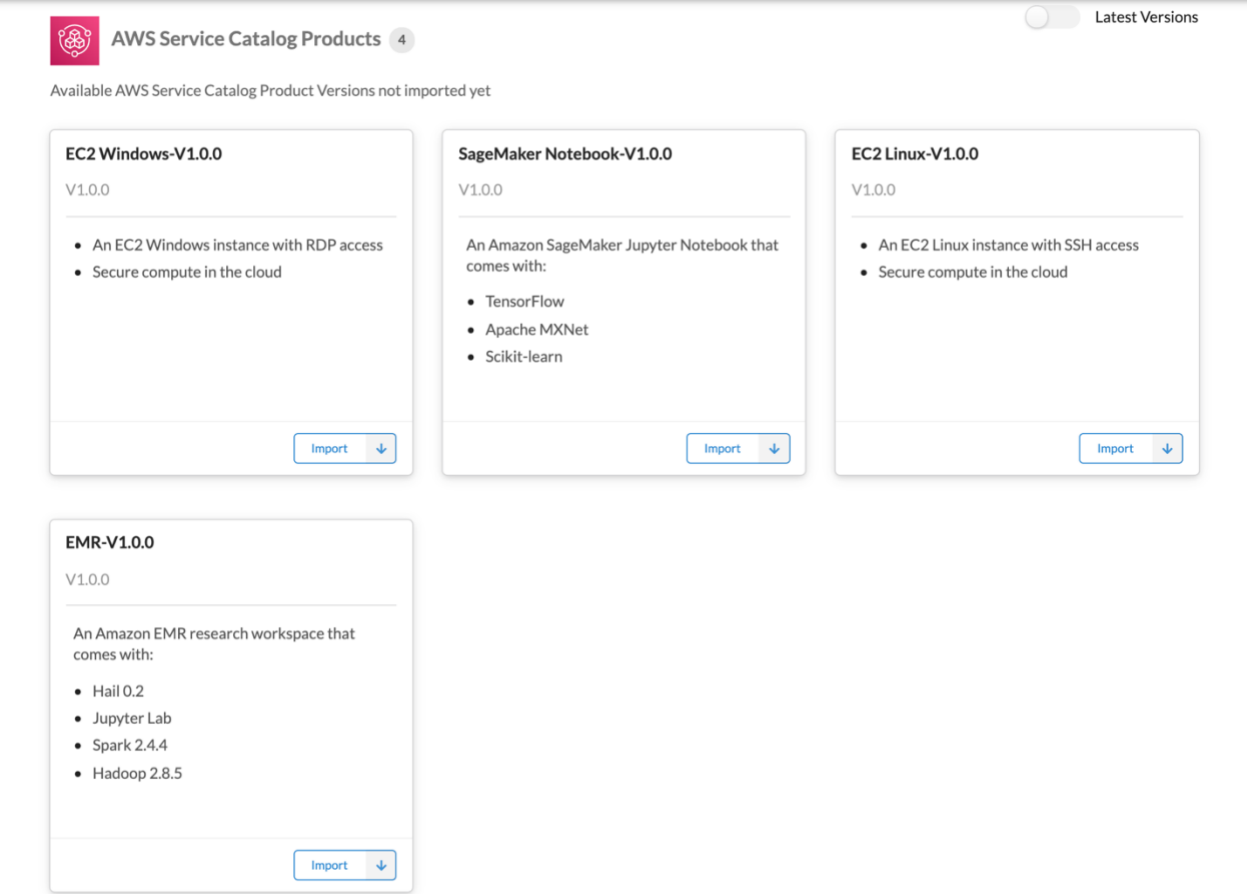


Figure: AWS Service Catalog Products

These four products come from the AWS Service Catalog portfolio created by the system during deployment. And they'll be ready for use once imported and configured.

If you wish to include other AWS computation resources in the future:

1. Add a new product to the existing Service Workbench portfolio in AWS Service Catalog
2. Update the role ServiceCatalogLaunchConstraintRole in [cloudformation.yml] to include permission needed to launch and terminate the product

Importing a workspace

In this section, the Amazon EC2 Linux is used as an example.

1. Choose the **Import** button under ec2-linux-instance.
2. Enter **Name** and **Description** so you can easily identify the workspace.

Configuring the workspace

Once you import a workspace type, perform the following actions:

1. Choose **Add Configuration**
2. Enter **ID**, **Name**, **Description**, and **Estimated Costs** for the configuration. A common naming convention here is to attach the instance size after the product name. For example, use ec2-linux-instance-V1-small for a small Linux Amazon EC2 instance.
3. Choose **Next**.
4. Add access control for the workspace configuration.
5. Choose **Next**.

The input parameters are parameters used for the product, AWS CloudFormation template. The number and type of parameters are different for different products. Most of the parameters used for the four system created products can be evaluated automatically at launch time. These parameters are available for selection in the drop-down when filling the input parameters page.

Configuration for EC2 Linux

For Amazon EC2 Linux, the only two fields that are not available in the drop-down are **InstanceType** and **AmiId**.

The following figures display screenshot images that exemplify Amazon EC2 Linux configurations.

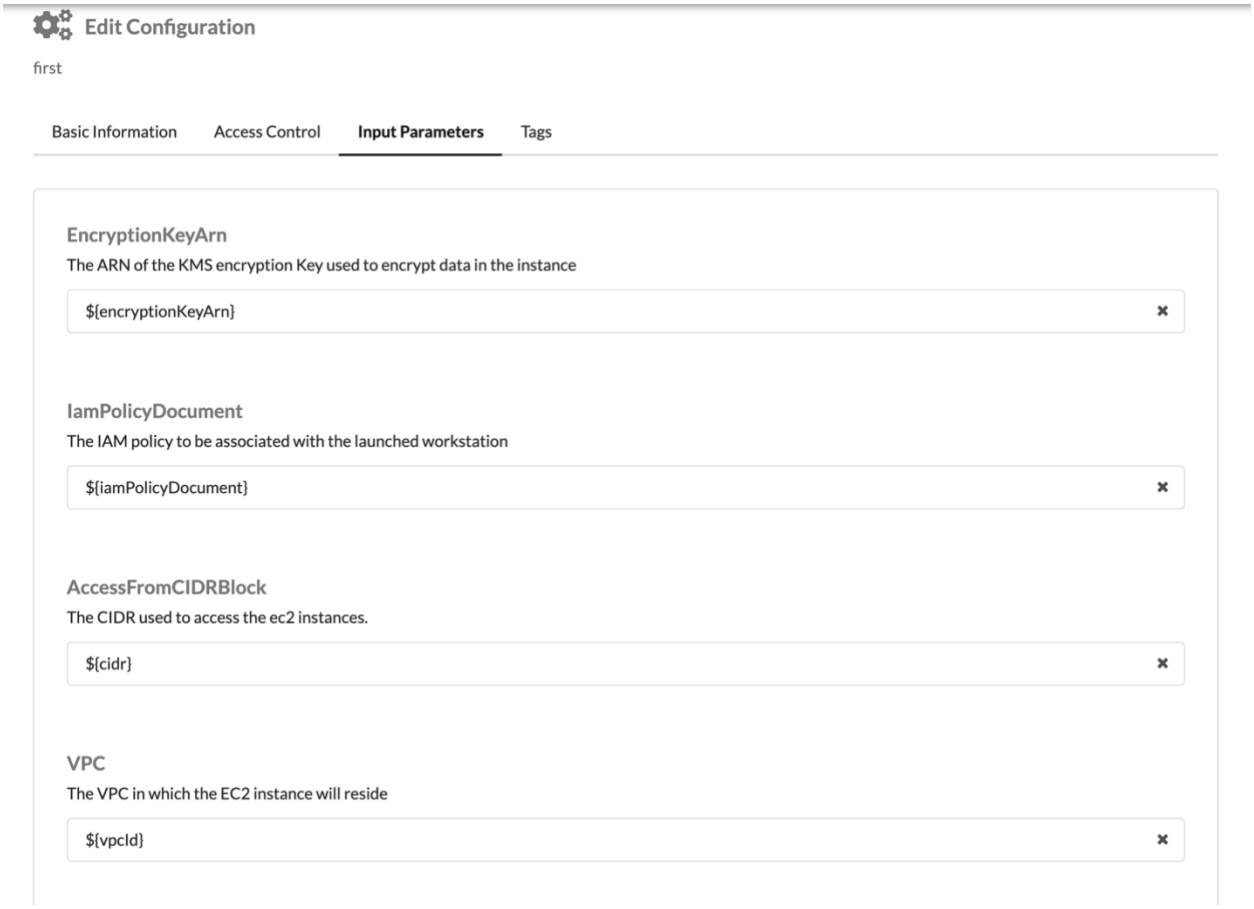


Figure: Configurations for Amazon EC2 Linux

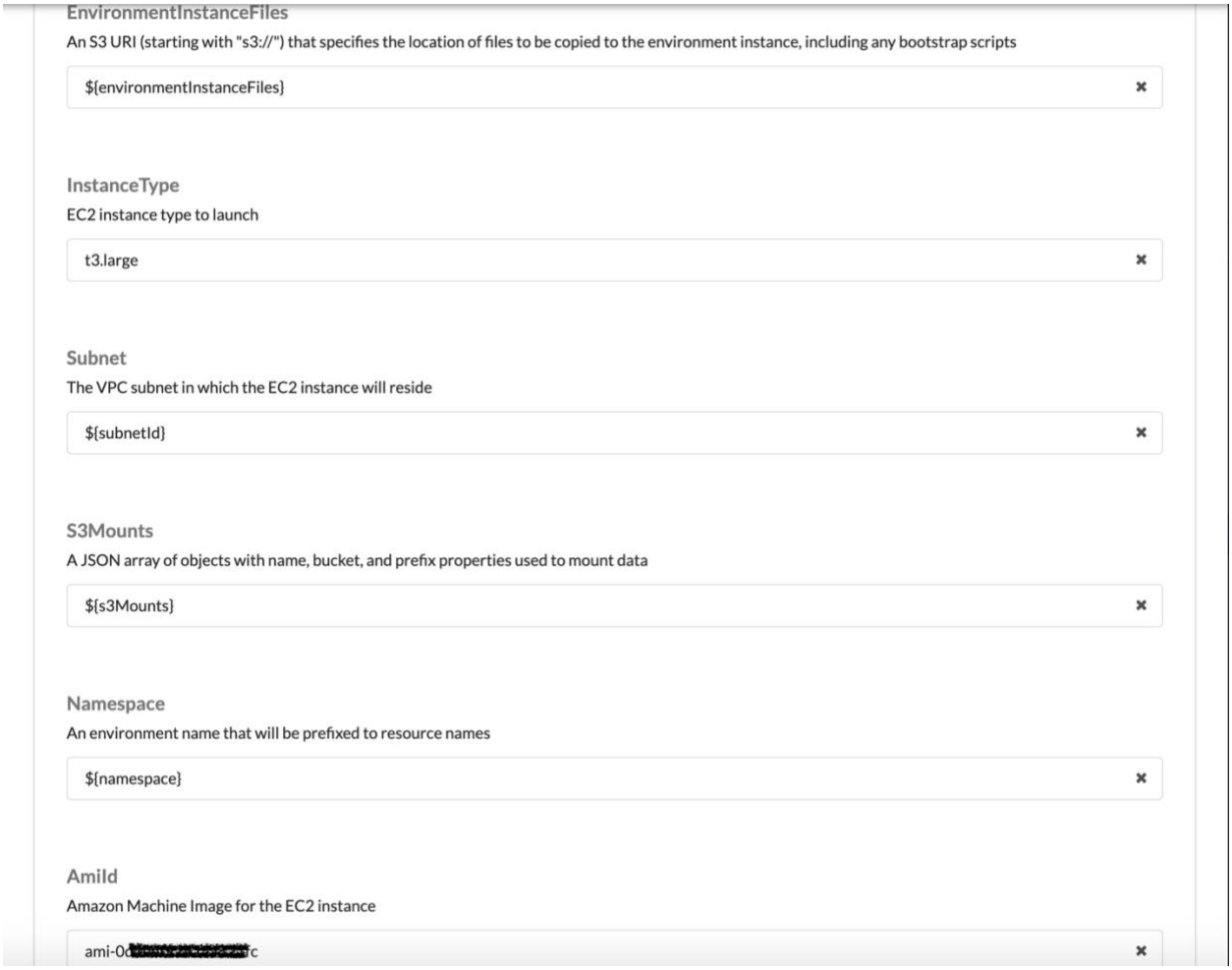


Figure: Configurations for Amazon Linux EC2

Configuration for Amazon EC2 Windows

For Amazon EC2 Windows, the only two fields that are not available in the drop-down are **InstanceType** and **AmiId**. (Use the AMI ID you copied in Prerequisites - AMI)

The following figures display screenshot images that exemplify Amazon EC2 Windows configurations.

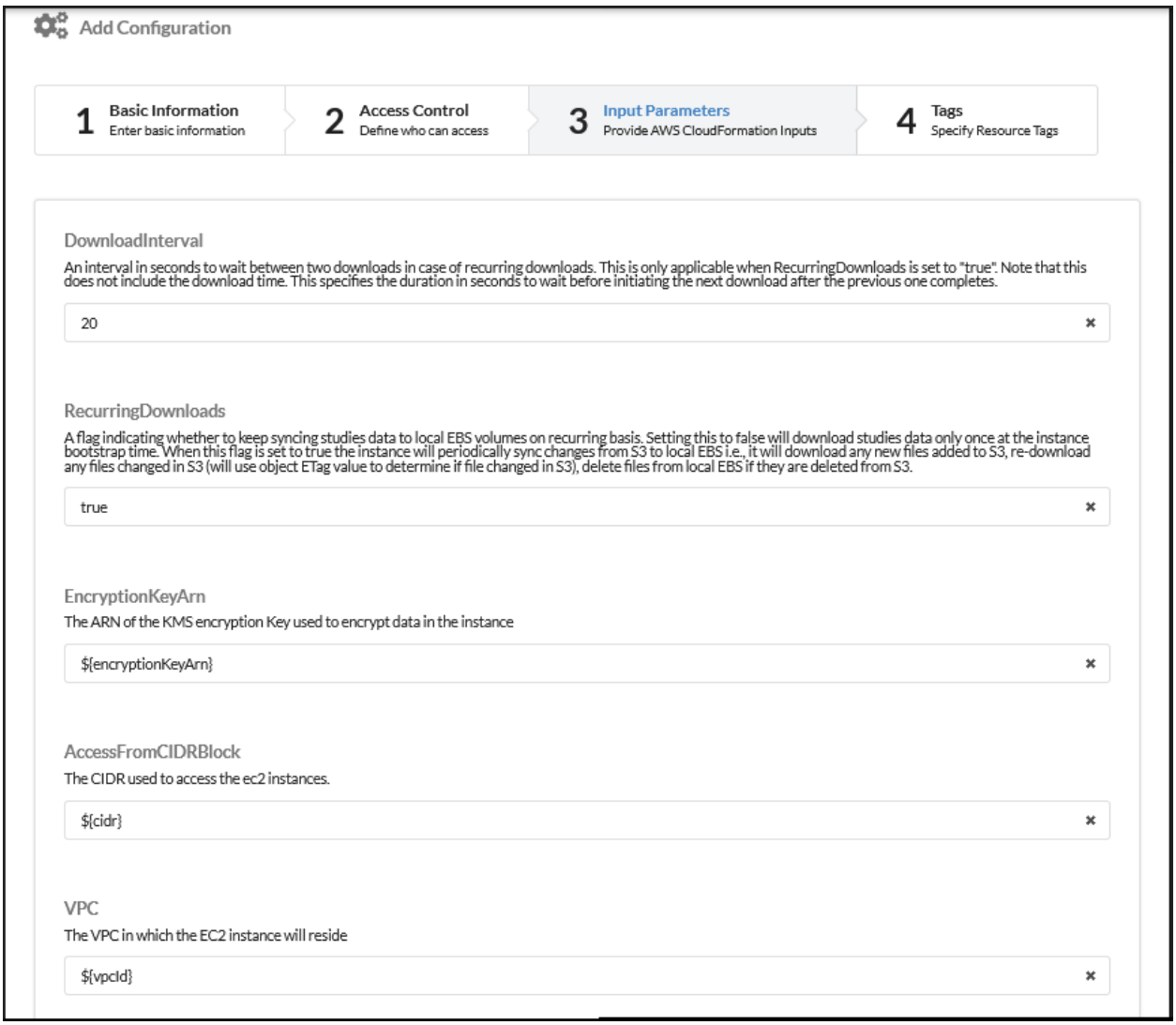


Figure: Configurations for EC2 Windows

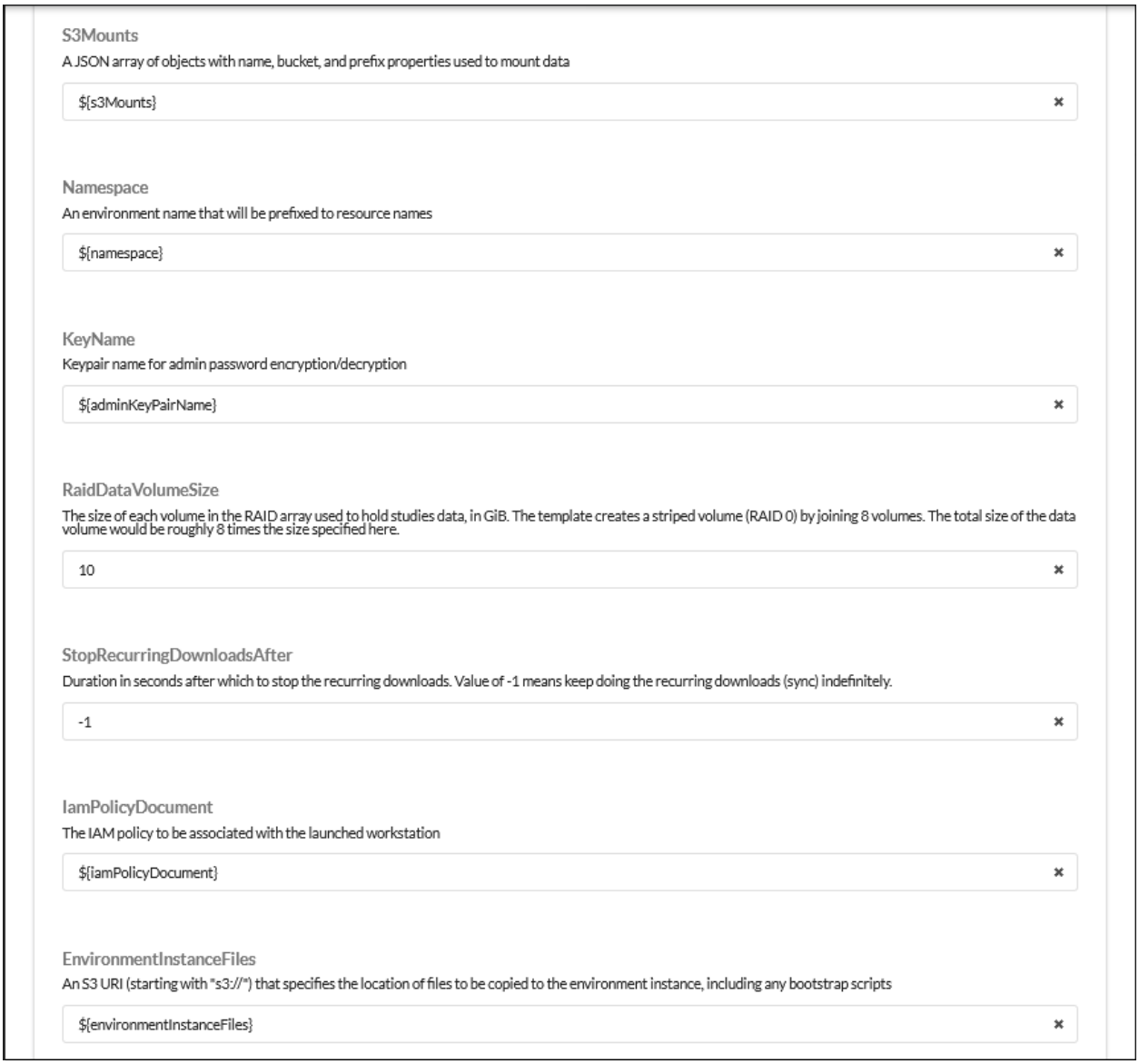


Figure: Configurations for EC2 Windows

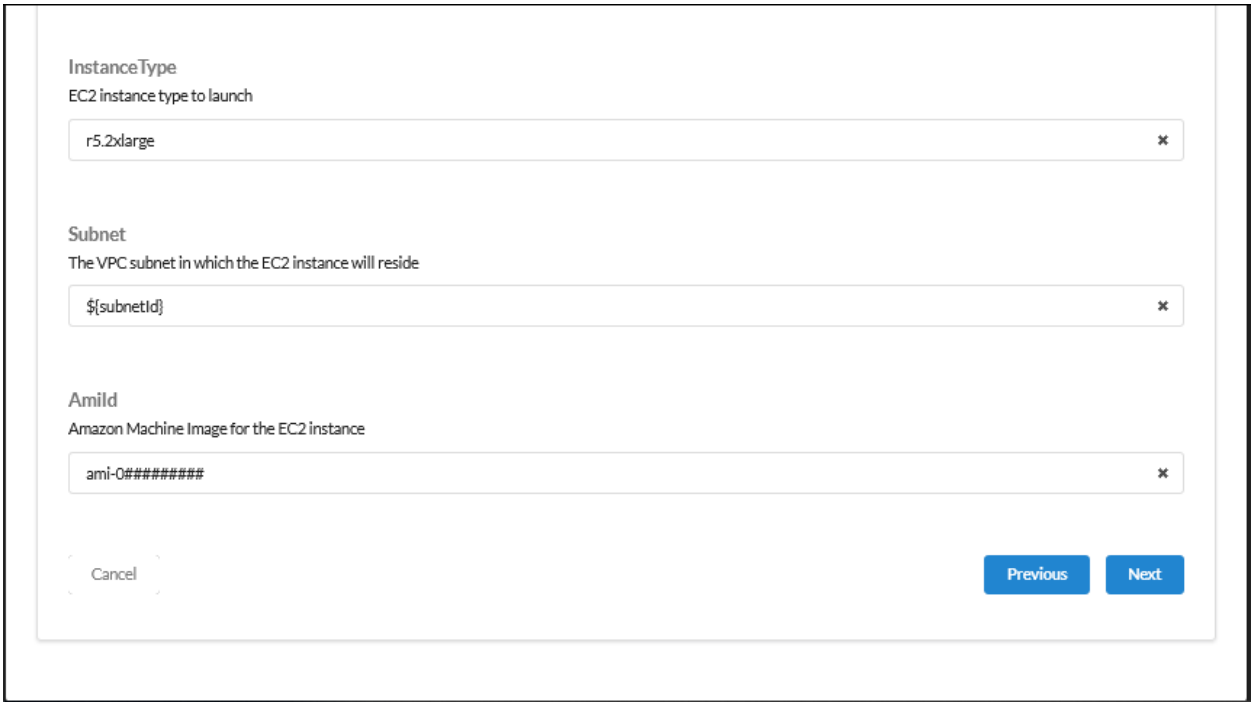


Figure: Configurations for EC2 Windows

Configuration for Amazon SageMaker

For Amazon SageMaker, the only field that's not available in the drop-down is '**InstanceType**'.

The following figures display screenshot images that exemplify Amazon SageMaker configurations.

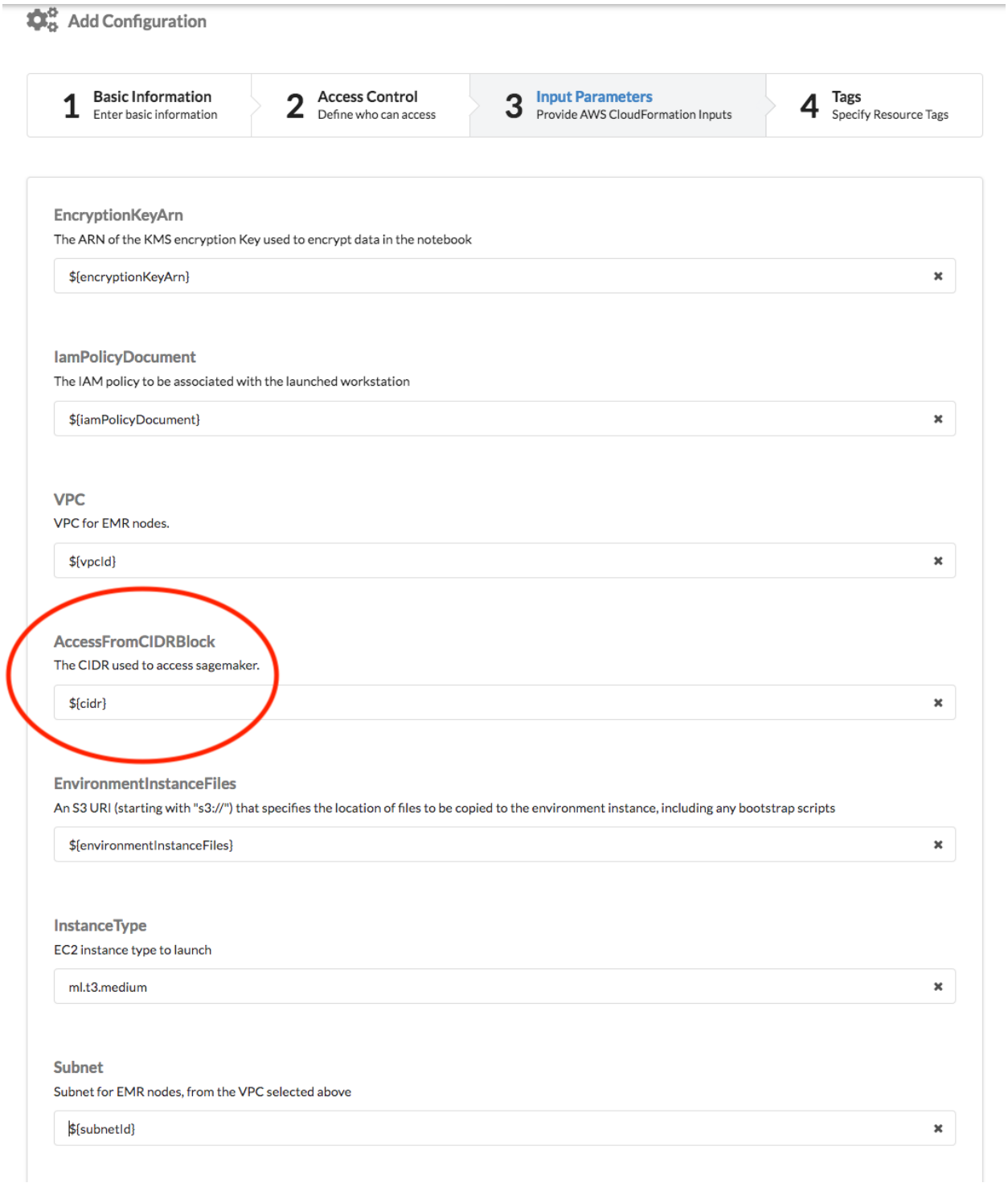


Figure: Configurations for Amazon SageMaker

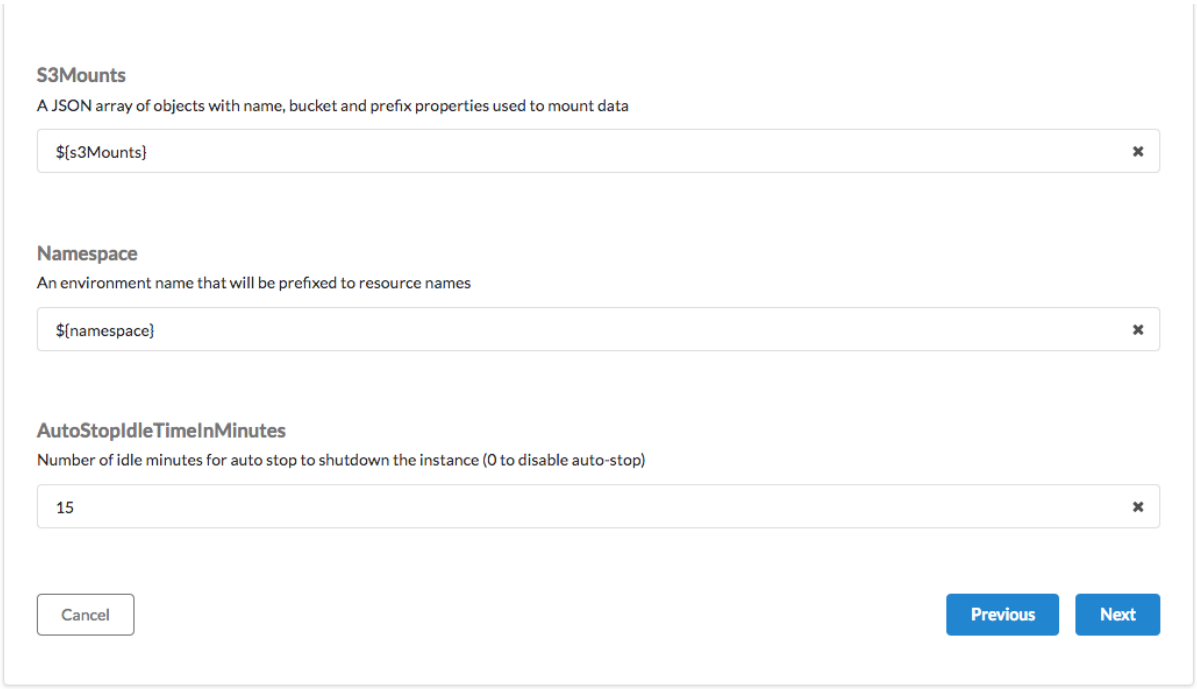


Figure: Configurations for Amazon SageMaker

Configuration for Amazon EMR

Note: EMR workspaces are not available if AppStream is enabled for the deployment. Therefore, it is unnecessary to import or add a configuration for this workspace.

Amazon EMR requires a few more fields that are not available in the drop-down menu, including the following:

- DiskSizeGB (>=10)
- CoreNodeCount (1-80)
- MasterInstanceType
- Market (ON_DEMAND / SPOT)
- WorkerBidPrice (only applicable when Market = SPOT. Specify 0 for Market = ON_DEMAND)
- WorkerInstanceType
- AmiId (Use the AMI id we copied in prerequisites - AMI)

The following figures display screenshot images that exemplify Amazon EMR configurations.

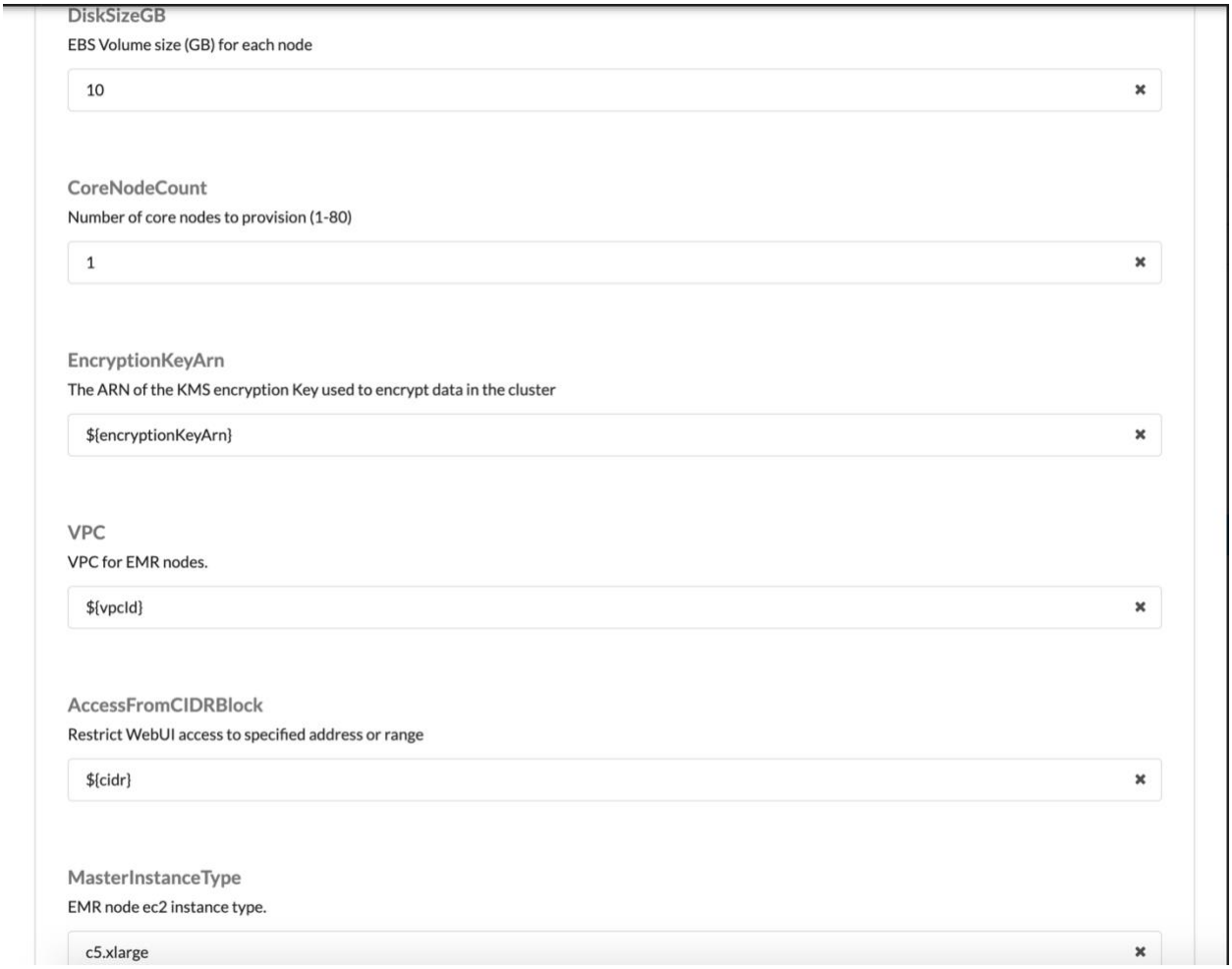


Figure: Configurations for Amazon EMR

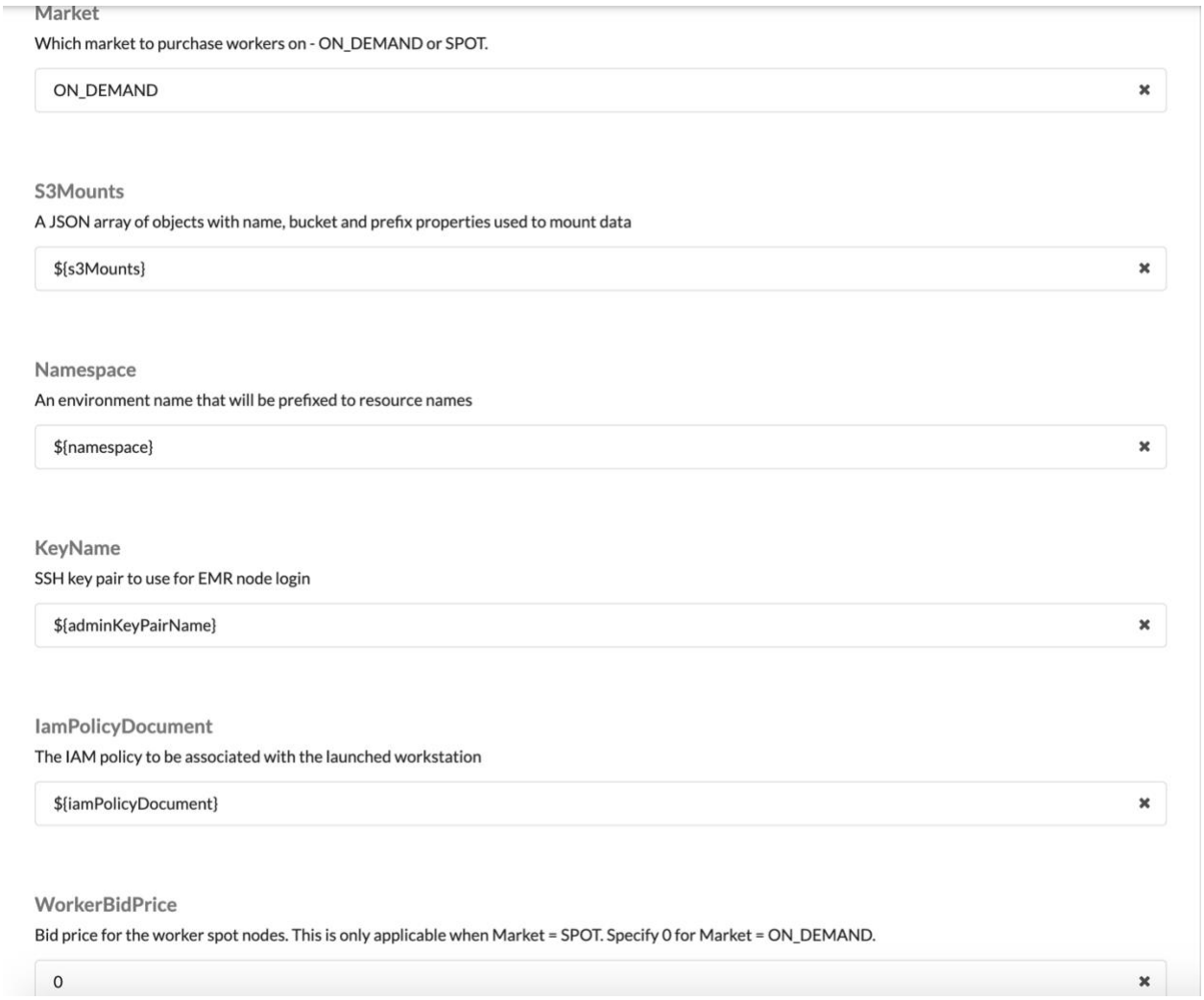


Figure: Configurations for Amazon EMR

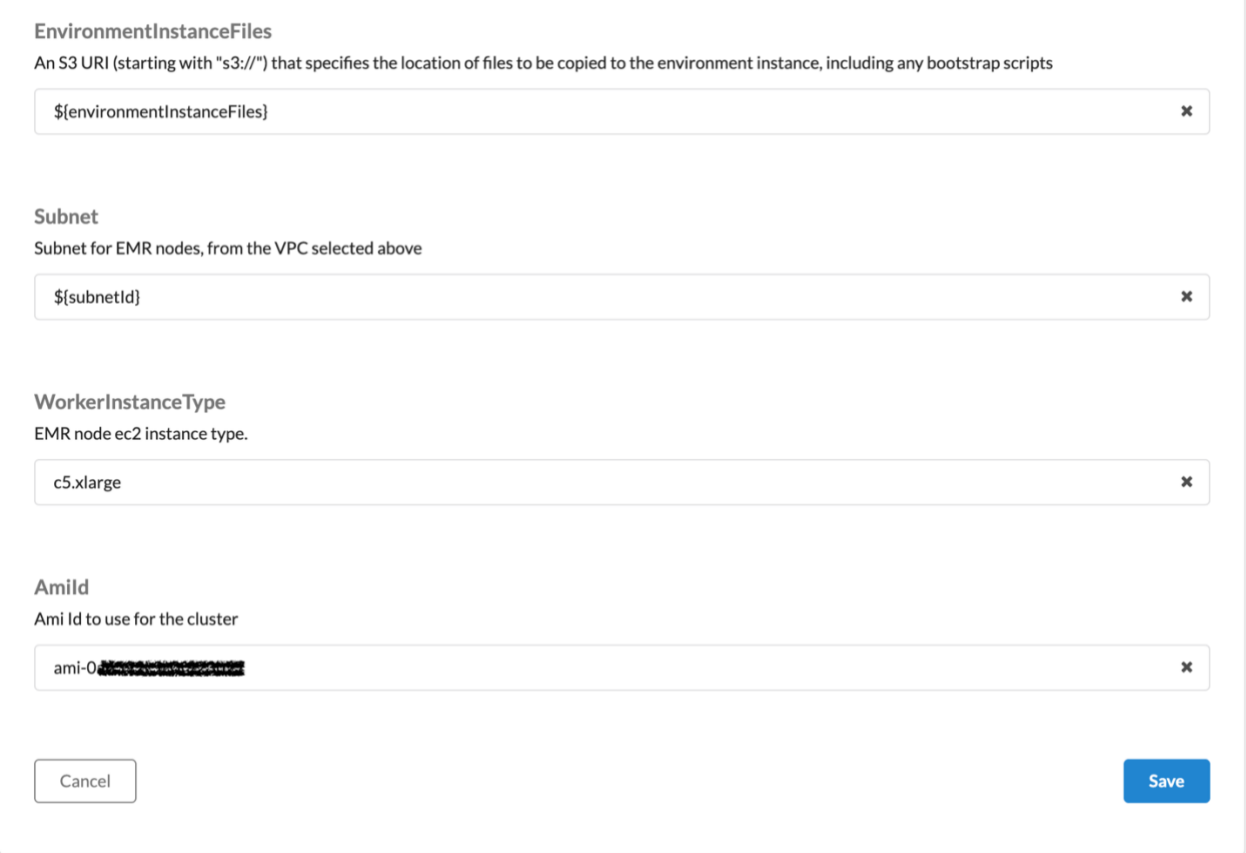


Figure: Configurations for Amazon EMR

Approving the workspace

Once the configuration completes, choose the **Approve** button; the newly created workspace type will be available for launch in the **Study and Workspace** tab.

Viewing logs

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/<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>.

Metrics

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

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

Service Workbench does not emit any custom metrics.

Deploying Updates

After a successful initial deployment, you can deploy individually to the five serverless projects that are a part of this solution.

Deploying Updates to the Infrastructure Serverless Project

```
$ cd solution/infrastructure$ pnpx sls deploy -s <stage>
```

Deploying Updates to the Backend Serverless Project

```
$ cd solution/backend$ pnpx sls deploy -s <stage>
```

Deploying Updates to the Machine-Images Serverless Project

```
$ cd solution/machine-images$ pnpx sls deploy -s <stage>
```

Deploying Updates to the Post-Deployment Serverless Project

```
$ cd solution/post-deployment$ pnpx sls invoke local -f postDeployment --env WEBPACK_ON=true -s <stage>
```

Deploying Updates to the UI Serverless Project

```
$ cd solution/ui$ pnpx sls package-ui --stage <stage> --local$ pnpx sls package-ui --stage <stage>$ pnpx sls deploy-ui --stage <stage> --invalidate-cache
```

Deploying Updates to the Prepare Devops Account Serverless Project

$ cd solution/prepare-devops-acc$ pnpx sls deploy -s <stage>

Reference

Service Workbench on AWS interacts with multiple AWS resources, including Amazon EC2, AWS IAM, AWS Organizations, and more. You can easily add an AWS IAM role to an Amazon EC2 instance, leverage our AWS Organizations with your account structure, and create multiple Amazon S3 buckets.

Adding an AWS IAM role to an Amazon EC2 instance

An Amazon EC2 instance can be assigned an instance profile that contains an AWS IAM role. The AWS IAM role gives the Amazon EC2 instance a set of permissions. The Amazon EC2 instance only performs the actions defined by its AWS IAM role. Adding an AWS IAM role to the Amazon EC2 instance allows your application to make API calls securely—eliminating the need to manage security credentials.

The Service Workbench deployment application must be able to create AWS resources. The easiest way to meet this requirement is to give the Amazon EC2 instance an administrator's role.

Adding administrator's role to a new Amazon EC2 instance

When creating a new Amazon EC2 instance for the Service Workbench deployment, an instance profile may be assigned to the Amazon EC2 instance. Choose **Create a new IAM role** located next to the **IAM role** drop-down.

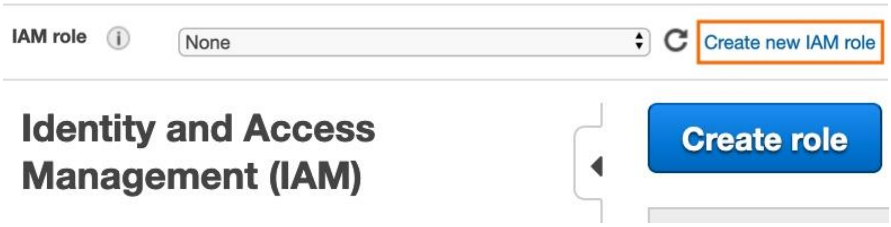


Figure: Create a new AWS IAM role

To continue the process, highlight Amazon EC2 and proceed to permissions.

1. Under **Permissions**, filter for **AdministratorAccess**.

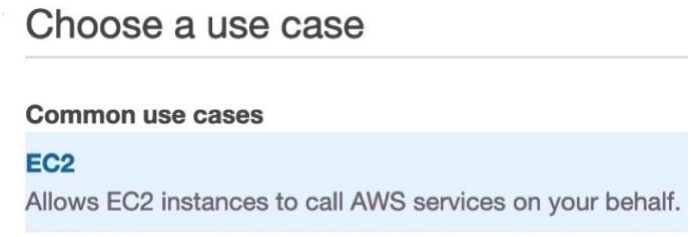


Figure: Permissions in Amazon EC2

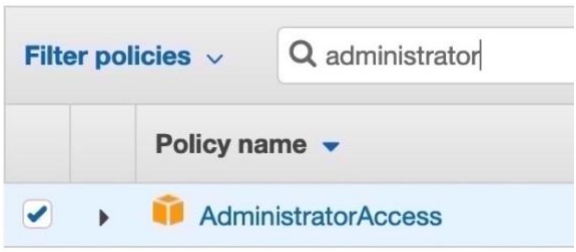


Figure: Filtering for AdministratorAccess

2. Proceed through **Tags**.
3. In the **Review** page, enter the role name.

Figure: Choosing a role name for an Amazon EC2 instance

4. In the **Amazon EC2** tab, refresh the AWS IAM role drop-down, and choose the administrator role that is to be attached to the new Amazon EC2 instance.

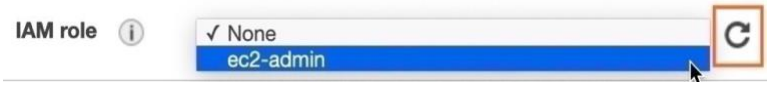


Figure: Selecting the administrator role of the Amazon EC2 instance

5. Now, proceed through the process to create an Amazon EC2 instance.

Adding a role to an existing instance

To add a role to an Amazon EC2 instance that is already running, select the Amazon EC2 instance in the EC2 Console. Open the **Action > Instance Settings** menu, and select **Attach/Replace IAM Role**.

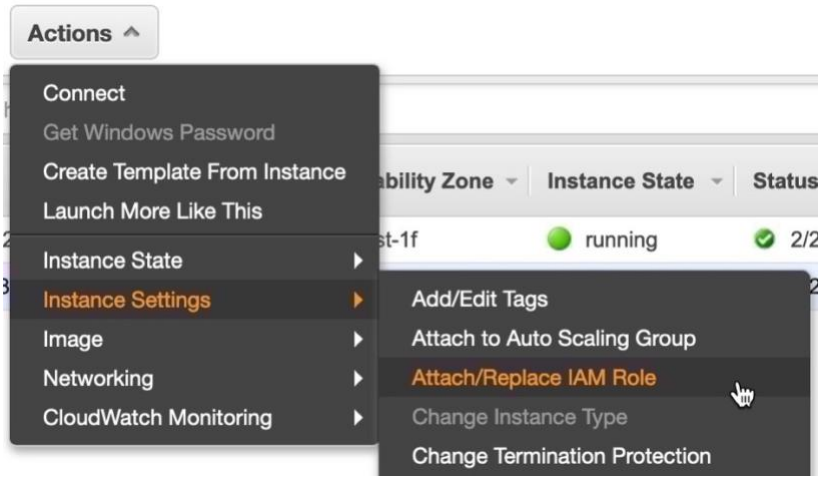


Figure: Attach/Replace an AWS IAM Role in the EC2 Console

In the **Attach/Replace IAM Role** screen, search for the role you created, select it, and choose **Apply**. **Figure 30** shows the screen where you can **Attach/Replace IAM Role**.

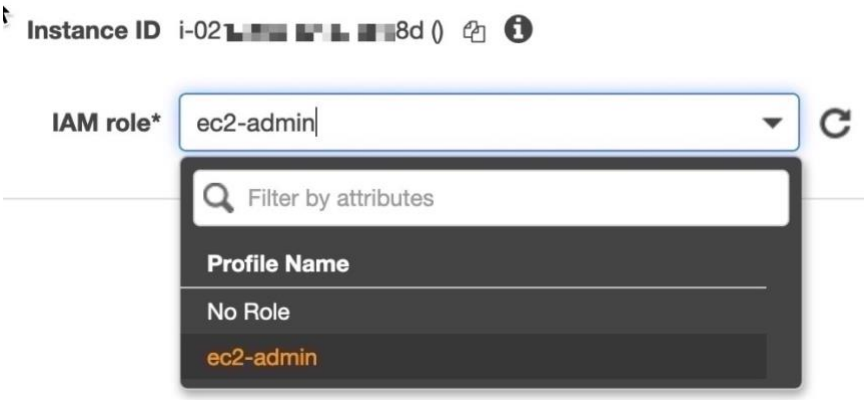


Figure : AWS IAM role search

Usage of AWS Cloud Services

This section describes some of the AWS Cloud services used by Service Workbench. The resource names usually include the **Namespace**, including the **Stage Name** used at deployment. You can deploy multiple instances of Service Workbench from the same account if you use a different Stage Name for each deployment.

Amazon EC2

Amazon EC2 is used only as a platform from which to deploy Service Workbench. For more details see the Deployment Instance section.

AWS IAM

Service Workbench creates several roles in your account. The role `<namespace>-prep-raas-master-MasterRole-XXX` is created when you run the Post Deployment SDC. This role possesses a trust relationship with the Main account from which you deployed Service Workbench. There are two polices that allow the Main account to assume a role in this Management account. The Account Structure defines each type of account. The following figure shows the AWS IAM '**Trust Relationships**' tab.

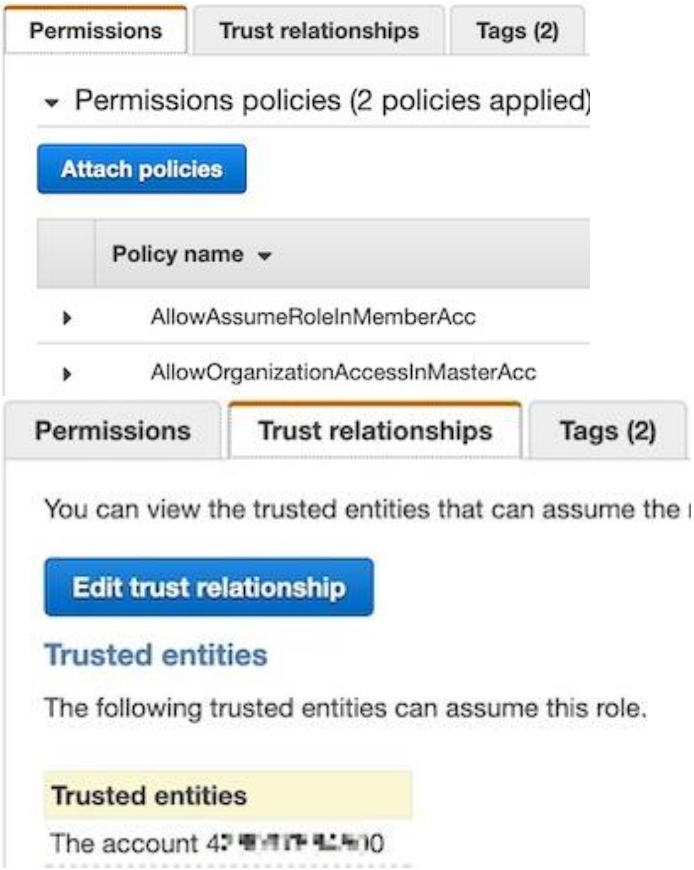


Figure: AWS IAM Trust Relationships Tab

An External ID is associated with the role. The External ID is an identifying string that is provided once a role is created. In order for the Trusted Entity (your Main account) to assume its role in the management account, it must supply this External ID. Providing the External ID of establishes a revocable relationship between the Trusted Entity and the Management account.

In the current Service Workbench deployment, the External ID is configured as a default value in the following string workbench:

```
main/solution/prepare-master-acc/config/settings/.defaults.yml
```

To change this value, create a stage-named configuration file (`mystagename.yml`) in the same directory. For more information, see the Configuration section. The following figure displays a screenshot image of the conditions that define how **Trusted Entities** assume a role.

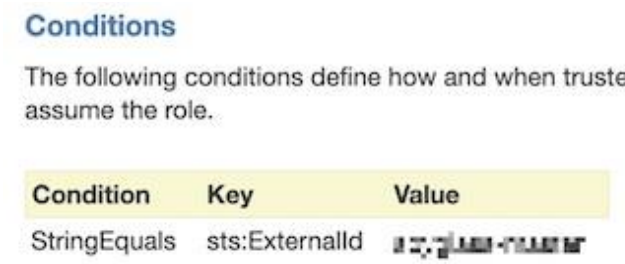


Figure: Defining Conditions for Trusted Entities

AWS Organizations

An AWS Organization is created in the **Management account**. The **Management account** is discussed in the Account Structure section of the Service Workbench User Guide . The AWS Organization use the **Management account** to create a separate account for each deployment. The account's name is the **Stage Name**used. The following figure shows a screenshot image of the AWS Organizations '**Accounts**' tab.

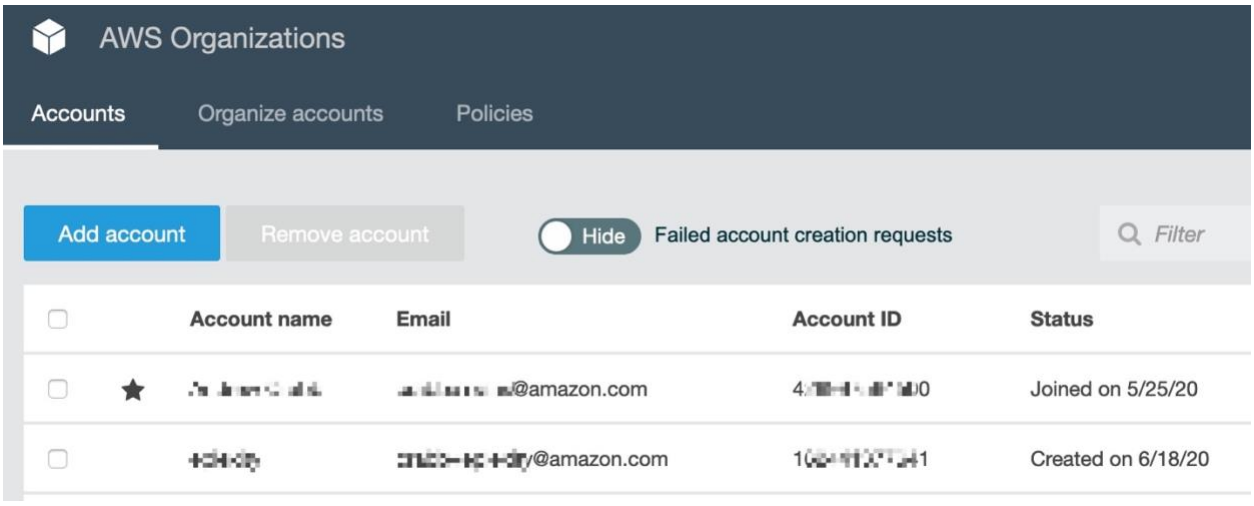


Figure: AWS Organizations Account Page

Amazon S3

Multiple Amazon S3 buckets are created by Service Workbench. Filtering by **Stage Name** shows the Amazon S3 buckets for a deployment. The following figure shows the Amazon S3 buckets for the Service Workbench deployment.

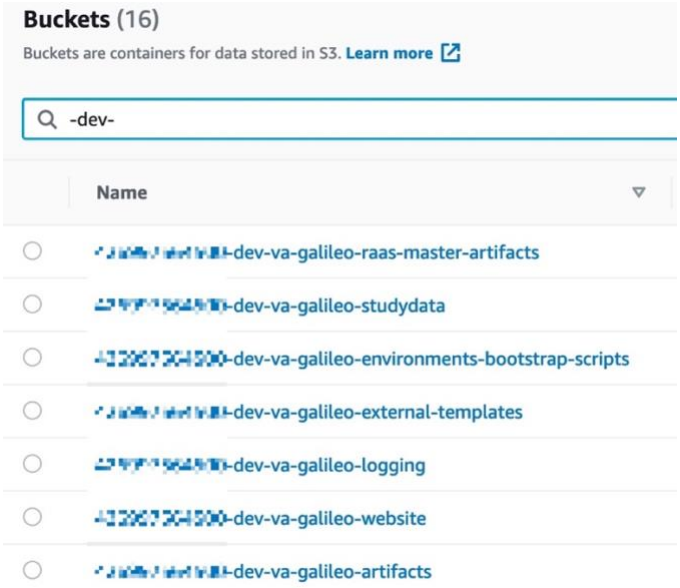


Figure: Amazon S3 Buckets for a Service Workbench Deployment

The '**studydata**' bucket contains all the data for the various studies in this deployment at the individual and organization level. The following figure displays an image of the contents within the studydata bucket.

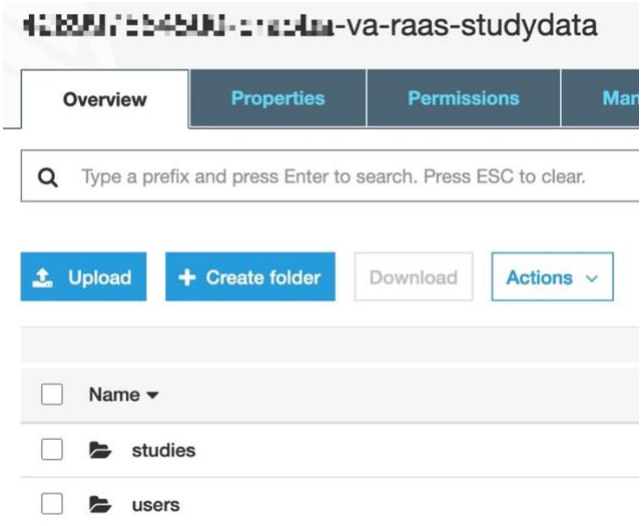


Figure: Amazon S3 StudyData Bucket

AWS Cost Explorer

Service Workbench has the ability to show actual cost incurred by workspaces running under the Management account. This is using the AWS Cost Explorer service in the AWS Management Console. AWS Cost Explorer must be manually set up for each Management account once in order to allow requests for cost data to process. Setting this up requires background processes to complete in the Management account, which can take up to 24 hours.