AWS Whitepaper

Setting Up Multi-User Environments in AWS (for Classroom Training and Research)

Setting Up Multi-User Environments in AWS (for Classroom Training and Research): AWS Whitepaper

Table of Contents

Setting Up Multi-User Environments in AWS (for Classroom Training and Research)

Publication date: **September 15, 2021** (*Document history*)

Amazon Web Services (AWS) can provide the ideal environment for classroom training and research. Educators can use AWS for student labs, training applications, individual IT environments, and cloud computing courses. This whitepaper provides an overview of how to create and manage multi-user environments in the AWS Cloud, so professors and researchers can leverage cloud computing in their projects.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

With AWS, you can requisition compute, storage, and other services on demand, gaining access to a suite of secure, scalable, and flexible IT infrastructure services as your organization needs them. This enables educators, academic researchers, and students to tap into the on-demand infrastructure of AWS to teach advanced courses, tackle research endeavors, and explore new projects – tasks that previously would have required expensive upfront and ongoing investments in infrastructure.

For more information, see Cloud Computing for Education and Cloud Products.

To access any AWS service, you need an AWS account. Each AWS account is typically associated with a payment instrument (credit card or invoicing). You can create an AWS account for any entity, such as a professor, student, class, department, or institution. When you create an AWS account, you can sign into the AWS Management Console and access a variety of AWS services.

Protect these security credentials and do not share them publicly. For more information, see AWS security credentials and AWS Management Console.

If you require more than one person to access your AWS account, AWS Identity and Access Management (IAM) enables you to create multiple users and manage the permissions for each of these users within your AWS account.

A user is a unique identity recognized by AWS services and applications. Similar to a user login in an operating system such Windows. macOS, or Linux, each user has a unique name and can identify themselves using various kinds of security credentials.

A user can be an individual, such as a student or teaching assistant, or an application, such as a research application, that requires access to AWS services. You can create users, groups, roles, and federation capabilities using the AWS Management Console, APIs, or a variety of AWS Partner products.

For instructions on how to create new users and manage AWS credentials, see Creating an IAM user in your AWS account in the *AWS Identity and Access Management User Guide*.

Depending on your teaching or research needs, there are several ways to set up a multi-user environment in the AWS Cloud. The following sections introduce three possible scenarios.

Topics

Scenario 1: Individual server environments

This scenario is excellent for labs and other class work that require users to access their own pre-provisioned Linux or Windows servers running in the AWS Cloud. The servers are running in Amazon Elastic Compute Cloud (Amazon EC2) instances.

The instances can be created by an administrator with a customized configuration that includes applications and the data needed to perform tasks for labs or assignments. This scenario is easy to set up and manage. Users in this scenario do not need their own AWS accounts, or credentials for any other servers. Since these users don't have an AWS account, they cannot allocate additional resources in the AWS Cloud.

Example

Consider a class with 25 students. The administrator creates 25 private keys and launches 25 Amazon EC2 instances—one instance for each student. The administrator shares the appropriate key or password with each student and provides instructions on how to log into their instance.

In this case, students do not have access to the AWS Management Console, AWS Command Line Interface, or AWS APIs, which prevents them from accessing other AWS services. Each student gets a unique private key (Linux) or sign-in credentials (Windows) along with the public hostname or IP address of the instance that they can use to log in.

Scenario 2: Limited user access to the AWS Management Console within a single account

This scenario is excellent for users that require control of AWS resources, such as students in cloud computing or high performance computing (HPC) classes. With this scenario, users are given restricted access to the AWS services through their IAM credentials.

Example

Consider a class with 25 students. The administrator creates 25 IAM users using the AWS Management Console, AWS Command Line Interface, or APIs, and provides each student with their

IAM credentials and a login URL for the AWS Management Console. The administrator also creates a permissions policy that can be attached to a user group or an individual user to allow or deny access to different services.

Each student (IAM user) has access to resources and services as defined by the access control policies set by the administrator. Students can log in to the AWS Management Console to access different AWS services as defined the policy. For example, they could launch Amazon EC2 instances and store objects in Amazon Simple Storage Service (Amazon S3) buckets.

Scenario 3: Separate AWS accounts for each user

This scenario, with optional consolidated billing, provides an excellent environment for users who need a completely separate account environment, such as researchers or graduate students. It is similar to Scenario 2, except that each IAM user is created in a separate AWS account, eliminating the risk of users affecting each other's services

Example

Consider a research lab with 10 graduate students. The administrator creates one management AWS account, which will own the AWS Organization. Then, the administrator provisions separate AWS accounts for each student within the AWS Organization. For each account, the administrator creates an IAM user in each of the accounts or manages the permissions through single sign-on users for each student and applies access control policies. Users receive access to an IAM user/role within their AWS account.

Users can log in to the AWS Management Console to launch and access different AWS services, subject to the access control policy applied to their account. Students don't see resources provisioned by other students, because each account is isolated from each other.

A key advantage of this scenario is that students can keep their accounts after the completion of the course. Each account can be set up as a standalone account, outside the AWS Organization. If the students have used AWS resources as part of a startup course, they can continue to use what they have built on AWS after the class, semester, or course is over.

Comparing the scenarios

The scenario you should select depends on your requirements. Table 1 provides a comparison of key features of these three scenarios.

Table 1: Comparison of scenarios

	Individual server environments	**Limited user access to AWS Management Console**	**Separate AWS account for each user**
Examples	Undergraduate labs	Graduate classes	Graduate research labs
Example uses	Labs or course work requiring a virtual server, AWS service, or separate applicati on instance	Courses in cloud computing or labs requiring variable resource needs (such as HPC)	Courses for startups, thesis, or research projects
Separate AWS accounts required for each user	No	No	Yes
Major steps for setup	Create and allocate Amazon EC2 resources and associated credentials	Create IAM users, create policies, and distribute credentials	Create separate member AWS accounts plus the steps in the Setting up Scenario 2: Limited user access to AWS Management Console section
Users can provision additional AWS resources, resulting in additional charges	No	Yes, depending on IAM services provided to users	Yes, depending on IAM services provided to users
Users have access to AWS Management Console or APIs	No	Yes	Yes

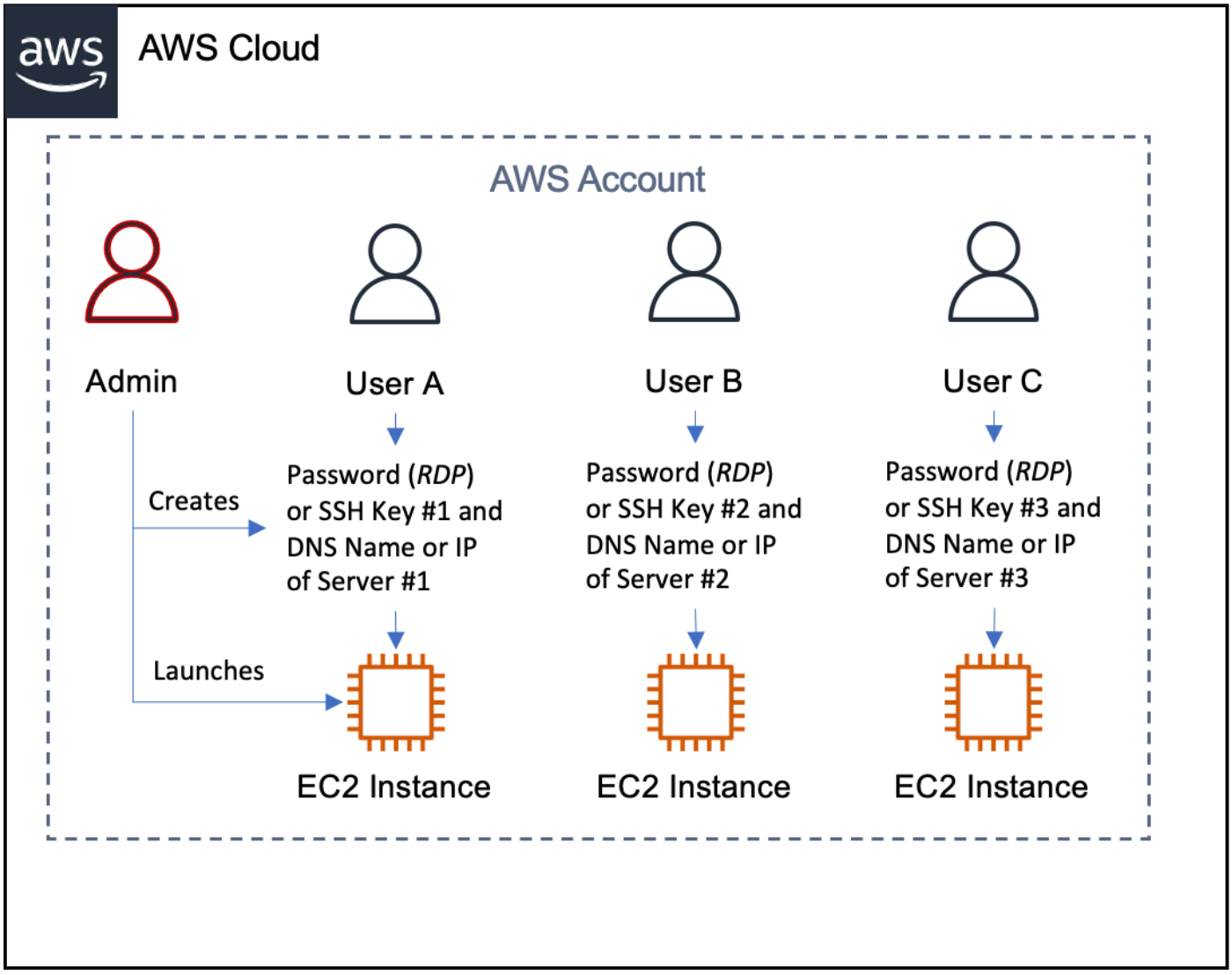
	Individual server environments	Limited user access to AWS Management Console	Separate AWS account for each user
User charges paid by the management AWS account	Yes	Yes	Yes, if consolidated billing is used
Separation between user environments	Yes, based on resource access configuration	Yes, if optional resource- based permissions are configured	Yes
Individual user credit cards or invoicing required	No	No	No, if consolidated billing is used
Billing alerts can be used to monitor charges	Yes	Yes	Yes

A large number of real-world use cases can benefit from implementing these scenarios. This section focuses on the education sector where multi-user, shared environments are required for setting up online classes, labs, and workshops for students. Both user and resource management are critical in these scenarios. Depending on your specific requirements, any of these scenarios can be used for setting up classrooms in the AWS Cloud. The following sections describe each of these scenarios in more detail.

Setting up Scenario 1: Individual server environments

With this scenario, users are provided access credentials to AWS resources. Users cannot access the AWS Management Console or launch new services. They receive the credentials to access specific AWS services that have already been launched by an administrator.

This scenario is a good match for simpler use cases in which users do not need to launch new AWS services. The following figure shows the architecture for this scenario.



Individual server environments

An administrator can give users their own unique SSH keys for Linux and password for Windows for security and separation between users. For labs that do not require security among users

(such as collaborative labs), the administrator can keep the keys or access credentials common for all the servers, and provide the unique access public DNS names of instances to the users. The administrator can choose the level of security and management appropriate for their needs.

Account setup

The administrator creates an AWS account for the user group. For example, this can be a shared account for a professor, class, department, or school. The administrator can also use an existing AWS account. New AWS account signup and access to existing AWS accounts is available on your Account page.

The administrator launches the required AWS services for each user, and provides resource access credentials to the users.

Cost tracking

If needed, the administrator tags the resources launched for different users. Cost allocation and resource tagging can help track usage by different users.

For more information, see Using Cost Allocation Tags in the AWS Billing and Cost Management and Cost Management documentation.

Monitoring resources

The administrator can set up AWS Budgets to monitor AWS resources. They can create billing alerts that automatically notify the designated recipient whenever the estimated AWS charges reach a specified threshold. The administrator can choose to receive an alert on the total AWS charges, or on charges for a specific AWS product or service. If the account has any limits, the administrator can use these as the threshold for receiving billing alerts.

For more information about setting up billing alerts with AWS Budgets, see Best practices for controlling access to AWS Budgets .

Reporting

Detailed usage reports are available for the administrator from the AWS Management Console. Reports are available for monthly charges and also for account activity in hourly increments.

For more information, see Detailed Billing Reports in the AWS Billing and Cost Management and Cost Management documentation.

Runtime environment

After the administrator provisions the account and launches the required AWS services, users can access their AWS resources using the provided credentials. For example, if Amazon EC2 instances are part of the class, users would be given keys or passwords to SSH (in Linux instances), or RDP (in Windows instances). Users would not have the credentials to log into the AWS Management Console, or to launch any new services.

Clean up the environment

When users have finished their work, or when the account limits are reached, the administrator can end the AWS resources. Because student users do not have their own AWS accounts, ending the launched services ensures that user work is deleted and no further charges are made.

Setting up scenario 2: Limited user access to AWS Management Console within a single account

For this scenario, the administrator creates IAM users and gives each user unique access credentials. With IAM, an administrator can securely control access to AWS services and resources.

The administrator can create and manage AWS users and groups, and use permissions to allow and deny access to AWS resources. Users can log into the AWS Management Console or AWS Command Line Interface and launch and access different AWS services, subject to the access control policies applied to their account. Users have direct control over the access credentials for their resources.

By default, when you create IAM users, they don't have access to any AWS resources. You must explicitly grant them permissions to access the resources that they need for their work. Permissions are rights that you grant to a user or group to let the user perform tasks in AWS. Permissions are attached to an IAM principal or an AWS Single Sign-On (SSO) permission set, and let the administrator specify what that user can do.

Depending on the context, administrators may be able to construct resource-level permissions for users that control the actions the user is allowed to take for specific resources (for example, limiting which instance the user is allowed to end).

For an overview of IAM permissions, see Controlling access to AWS resources using policies in the AWS Identity and Access Management documentation, and read Resource-Level Permissions for EC2–Controlling Management Access on Specific Instances on the AWS Security Blog.

To define permissions, administrators use *policies*, which are documents in JSON format. A policy consists of one or more statements, each of which describes one set of permissions. Policies can be attached to IAM users, groups, or roles. AWS Policy Generator is a handy tool that lets administrators create policies easily.

For example policies that are relevant to multi-user environments, see Appendix B: Example IAM user policies.

For more information about policies, see Policies and permissions in IAM.

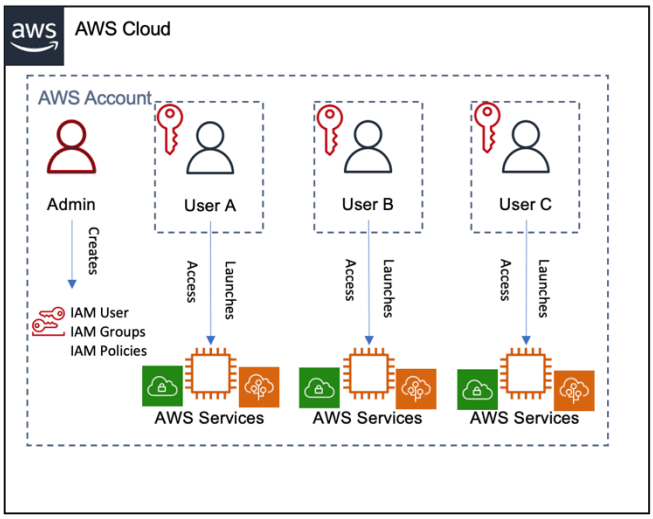
A useful option in this scenario is for the administrator to tag resources and write appropriate resource-level permissions to limit IAM users to specific actions and resources. A tag is a label

you assign to an AWS resource. For services that support tagging, apply tags using the AWS Management Console, AWS Command Line Interface, or API requests. This approach enables fine-grained control to which resources a user can access, and what actions they can take on those resources.

The administrator will also need to write policies to prevent users from manipulating the resource tags. For example, for Amazon EC2 tags, the administrator should disable the `ec2:CreateTags` and `ec2:DeleteTags` actions.

This scenario is also good for use cases that require collaboration among users. As described previously, a user can give other IAM users access to specific actions on their resources using a mix of user-level and resource-level permissions. A good example is a collaborative research project where students allow other members of their team access to software in their Amazon EC2 instances and data stored in their Amazon S3 buckets.

This scenario can be useful when the users need to access the AWS Management Console, launch new services, interact with services for complicated cloud-based application architectures, or exercise more control over accessing and sharing resources. The following figure shows the architecture for this scenario.



Limited user access to AWS Management Console

As shown in the preceding figure, this scenario works well with a single AWS account. The administrator needs to create IAM users and groups to apply access control policies for the environment. Example IAM user policies for setting up this scenario are described in Appendix B: Example IAM user policies.

Account setup

The administrator creates one AWS account for the group. For example, this can be a shared account for a professor, class, department, or school. An existing AWS account can also be used. New AWS account signup and access to existing AWS accounts is available on the Account page.

The administrator then creates an IAM user for each user with the AWS Management Console, AWS Command Line Interface, or AWS API. These IAM users can belong to one or more IAM groups within a single AWS account.

Alternatively, the administrator can deploy SSO and create an SSO user for each student, teaching assistant, or professor. This approach allows users to log into the account through federation. Each SSO user can have one or more permission sets assigned to them, depending on the role that they need to assume to log into the account.

Based on environment requirements, the administrator attaches custom policies to IAM users or IAM groups to restrict certain AWS resources that can be launched and used. Thus, users can only launch AWS services for which permissions have been granted. Users are provided credentials for their IAM user, which can be used to log in to the AWS Management Console, access AWS services, and call APIs.

Information required for account setup

To create an account and set up IAM-based access control, an administrator needs the following information:

- An AWS account for the group. This account could belong to the school, department, or professor. If no account exists, a new account must be created.
- Name and email address of the user.
- Required AWS resources and services and the operations permitted on them. This is required to determine the access control policies to be applied to each IAM user.
- Contact information for the billing reports and alerts.
- Contact information for the usage reports and alerts.

Providing access to users

With SSO, the administrator can use the sample IAM policies from Appendix B to create custom permission sets to assign to each group of users using the IAM user policies.

Next, the administrator needs to create an AWS SSO user for each of the students, and assign each user to the relevant permission set. Students then can log in using the AWS SSO Sign-in URL. See this Basic AWS SSO Configuration video tutorial.

For basic instructions on how to add IAM user policies, see Appendix A: Adding IAM user policies.

For example, IAM user policies for setting up this scenario, see Appendix B: Example IAM user policies.

If the administrator decides not to use SSO, they can add IAM users with roles and custom policies to the AWS account directly, to implement required access control logic for the different kinds of users in the group. The administrator then provides the IAM user login information to the corresponding members of the group.

Cost tracking

All users can tag their resources for services with tagging capability. With the cost allocation feature of AWS Account Billing, the administrator can track AWS costs for each user.

For more information, see Using Cost Allocation Tags in the AWS Account Billing documentation.

Monitoring resources

AWS Budgets can help monitor AWS resources. Billing alerts automatically notify users whenever the estimated charges on their current AWS bill reach a threshold they define. Users can choose to receive an alert on their total AWS charges or charges for a specific AWS product or service. If the account has any limits, the administrator can use these as the threshold for sending billing alerts.

For more information about setting up billing alerts with AWS Budgets, see Best practices for controlling access to AWS Budgets.

Reporting

Detailed usage reports are available for the administrator from the AWS Management Console. Reports are available for monthly charges and also for account activity in hourly increments.

For more information, see Detailed Billing Reports.

Runtime environment

Users can log into the AWS Management Console (as an IAM user or with an AWS SSO user) with the login information provided to them by the administrator. They can launch and use resources defined by the rules and policies set by the administrator. For example, if they have the appropriate permissions, they can launch new Amazon EC2 instances or create new Amazon S3 buckets, upload data to them, and share them with others.

An IAM user might be granted access to create a resource, but the user's permissions, even for that resource, are limited to what's been explicitly granted by the administrator. The administrator can also revoke the user's permission at any time. Setting proper resource and user-based permissions helps prevent an IAM user from taking actions on resources belonging to other IAM users in the AWS account. For example, an IAM user can be prevented from terminating instances belonging to other IAM users in the AWS account.

For more information, see Controlling access to AWS resources using policies.

Clean up the environment

When users have finished their work or when the account limits are reached, they (or the administrator) can end the AWS resources. Administrators can also delete the IAM users. If an instance of SSO was created for the users to log in, the directory should be disabled. The users will lose their work unless they take steps to save it (a procedure that is beyond the scope of this whitepaper).

Setting up Scenario 3: Separate AWS account for each user

In this scenario, an administrator creates separate AWS accounts for each user who needs a new AWS account. These accounts can optionally be combined into a single AWS Organization, and a single AWS account can be designated as the management account, using AWS Organizations. After the student accounts become members of the Organization, the management account can become the payer account and all the accounts can benefit from consolidated billing, which provides a single bill for multiple AWS accounts. The administrator then creates an IAM user in each AWS account and applies an access control policy to each user. Users are given access to the IAM user within their AWS account, but do not have access to the AWS account root user.

The administrator should deploy SSO in the management account to create users to grant access to each account through federation centrally. This allows the accounts to be managed by an administrator, consistent with the required policies for the user environment.

Users can log into the AWS Management Console with their IAM credentials and then launch and access different AWS services, subject to the access control policies applied to their account. Since students have access to their individual accounts, they have direct control over the access credentials for their resources (such as the creation and deletion of SSH keys), and they can also share these resources with other users and accounts as needed.

This scenario is good for setting up collaborative multi-user work environments. To implement it, users can create an IAM role, which is an entity that includes permissions, that isn't associated with a specific user. Users from other accounts can then assume the role and access resources according to the permissions assigned to the role.

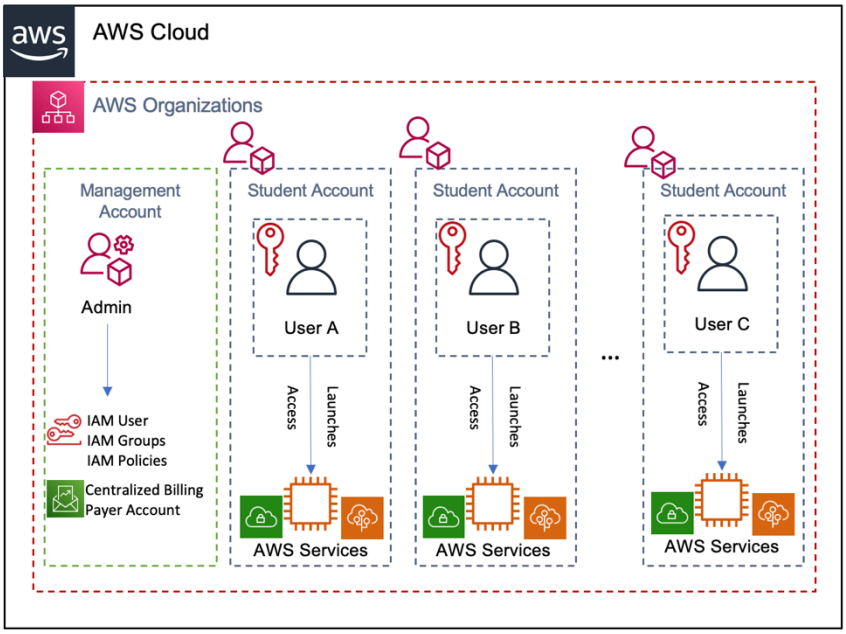
For more information, see Roles terms and concepts.

This scenario offers maximum flexibility for users and is helpful when they need to access the AWS Management Console to launch new services. TIt also gives users flexibility in working with complicated cloud-based application architectures and more control over accessing and sharing their resources.

Having separate AWS accounts for each user works well for both short-term and long-term usage. For short-term usage, AWS resources, IAM users, and even AWS accounts can be terminated after the work is done. For long-term usage, the AWS accounts for some or all users are kept alive at the end of the current engagement. All work done can be easily preserved for future use.

Users can also be provided full administrator access to their AWS account (besides the IAM-based access they initially had) to continue their work. For example, consider an entrepreneurship class where some students might develop new solutions or intellectual property using AWS resources that they want to retain for future use or for immediate deployment. Their work can be easily turned over to them by giving them full access to their AWS account.

Another benefit of this scenario is that in the AWS Management Console, users cannot see resources belonging to any other users in the group, since each user is working from their own AWS account. The following figure shows the architecture for this scenario.



Separate AWS account for each user

Account setup

The administrator deploys AWS Organizations in the management accounts, then provisions an AWS account for each user in the group. Independent AWS accounts (with unique AWS account IDs) are created for each user.

The administrator creates the accounts using the AWS Organizations `CreateAccount` API, which creates an account that requires the credentials of the AWS account root user to be reset.

In the management account, the administrator deploys AWS SSO, and creates an AWS SSO user for each of users that requires access to the AWS accounts. Based on the environment requirements,

the AWS SSO users are assigned to their relevant AWS SSO permission sets, that are customized with IAM policies, allowing each user to only use the services for which permissions have been granted.

Alternatively, the administrator can create an IAM user in each user's AWS account. Based on environment requirements, custom policies are attached to IAM users individually to constrain AWS resources that can be launched and used. Users can only launch AWS services for which permissions have been granted.

Users are provided credentials to log into the AWS Management Console, access AWS services, and call APIs. Users do not have access to the root user credentials of the AWS account and cannot change the IAM access policies enforced on the account.

Using AWS Organizations, an administrator can set up consolidated billing for the group. Consolidated billing (offered at no additional charge by AWS) enables consolidation of payment for multiple AWS accounts by designating a single payer account, the management account within the AWS Organization. Consolidated billing provides a combined view of AWS charges incurred by all accounts as well as a detailed cost report for each individual member AWS account associated with the management account.

For detailed information about how to set up consolidated billing, see Consolidated billing for AWS Organizations.

Another benefit from this scenario is that the administrator can set controls across every account using service control policies (SCPs) to restrict access to specific resources and services independently of the user's permissions.

Information required for account setup

The following information is required for creating accounts and setting up IAM-based access control:

- AWS management account for the group. This account could belong to the school, department, or professor. If no account exists, a new account is created. This account is necessary for setting up AWS Organizations and consolidated billing.
- Name and email addresses of users.
- AWS account credentials for users who have existing AWS accounts that they want to use in this environment. These accounts will join the AWS Organization users who do not have an AWS

account or do not want to use their existing account will need new accounts provisioned for them.

- Required AWS resources and services and the operations permitted on them. This is required to determine the access control policies to be applied to each IAM user.
- Contact information for the billing reports and alerts.
- Contact information for the usage reports and alerts.

Providing access to users

Using SSO, the administrator creates different permissions sets, which are custom IAM policies that can be used to grant resource access to the accounts within the AWS Organization to a specific user. Then the administrator creates an associated SSO user, assigns it to the user's account, and attaches a permission set to that user, based on the level of permissions that the user needs.

In this case, there could be a permission set for an administrator, a permission set for the teaching assistant, and a permission set for the students. Changes to the permissions will apply immediately to all the users using the same permission set in their account. Finally, the administrator can generate login credentials for each user, so they can access the accounts each user has access to through the SSO portal.

For more information on how to manage access to your accounts and assign policies to your permission sets, see Manage SSO to your AWS Accounts. See this SSO Configuration tutorial video to understand AWS SSO better.

Alternatively, the administrator can add IAM users with roles and custom policies for each user in each AWS account to implement required access control logic for the different types of users in the group. Login information for the IAM users is provided to the corresponding users in the group.

For basic instructions on how to add IAM user policies, see Appendix A: Adding IAM user policies.

For an example of setting up IAM user policies for this scenario, see Appendix B: Example IAM user policies.

Cost tracking

Consolidated billing makes it easy to track AWS costs because it shows the administrator a combined view of charges incurred by all AWS accounts, as well as a detailed cost report for

each individual AWS account within the organization. Consolidated billing is included with AWS Organizations, where the management account pays the charges of all member accounts.

All users can also tag their resources for services with tagging capability. An administrator can then use the cost allocation feature of AWS Account Billing to track AWS costs for each user.

For more information, see Using Cost Allocation Tags and Viewing your bill in the AWS Billing and Cost Management and Cost Management documentation.

Monitoring resources

AWS Budgets alerts can help monitor AWS resources. Billing alerts can automatically notify users whenever the estimated charges on their current AWS bill reach a threshold they define. Users can choose to receive an alert on their total AWS charges or charges for a specific AWS product or service. If the account has any limits, the administrator can use these as the threshold for sending billing alerts.

For more information about setting up billing alerts with AWS Budgets, see Best practices for controlling access to AWS Budgets.

Reporting

Detailed usage reports are available for administrators from the AWS Management Console. Reports are available for monthly charges as well as for account activity in hourly increments.

For more information, see Detailed Billing Reports.

Runtime environment

Users can log into the AWS Management Console as an IAM user with the login information provided to them by the administrator. They can launch and use resources defined by the rules and policies set by the administrator. For example, if they have the appropriate permissions, users can launch new Amazon EC2 instances or create new Amazon S3 buckets, upload data to them, and share them with others.

Because accounts are independent, each user sees only their own AWS resources in the AWS Management Console.

Clean up the environment

When the users have finished their work or when the account limits are reached, they (or the administrator) can optionally terminate the AWS services. The administrator can also delete the IAM users or the SSO users and revoke the access to the account.

When the account is no longer in use, it can be closed, ending all the resources within the account.

Keeping accounts alive

If the users want to retain their AWS accounts, they can request the root user account credentials from their administrator. The administrator would remove the account from the Organization and users would need to provide their own billing information. The users will get login and security credentials to their AWS account.

Conclusion

Multi-user, shared environments with custom access control policies are a common use case for AWS customers. Typical requirements include both user and resource management to allow controlled access to AWS resources for multiple users. This whitepaper presented three scenarios that covered a wide array of use cases with these requirements.

- Scenario 1: Individual server environments provides access to customized work environments on AWS and is suitable for use cases like undergraduate labs.
- Scenario 2: Limited user access to the AWS Management Console within a single account provides IAM user access to users from a single AWS account suitable for use cases like graduate classes.
- Scenario 3: Separate AWS accounts for each user provides independent AWS accounts for each user (with consolidate billing), which is suitable for graduate research and entrepreneurship courses.

In this whitepaper, we focused on the short- to medium-term education and research environments as the example domain, but the same or similar scenarios may also be implemented for other use cases.

Contributors

Contributors to this document include:

- KD Singh, Amazon Web Services
- Leo Zhadanovsky, Chief Technologist Education, Amazon Web Services
- Alex Torres, Solutions Developer, Amazon Web Services

Further reading

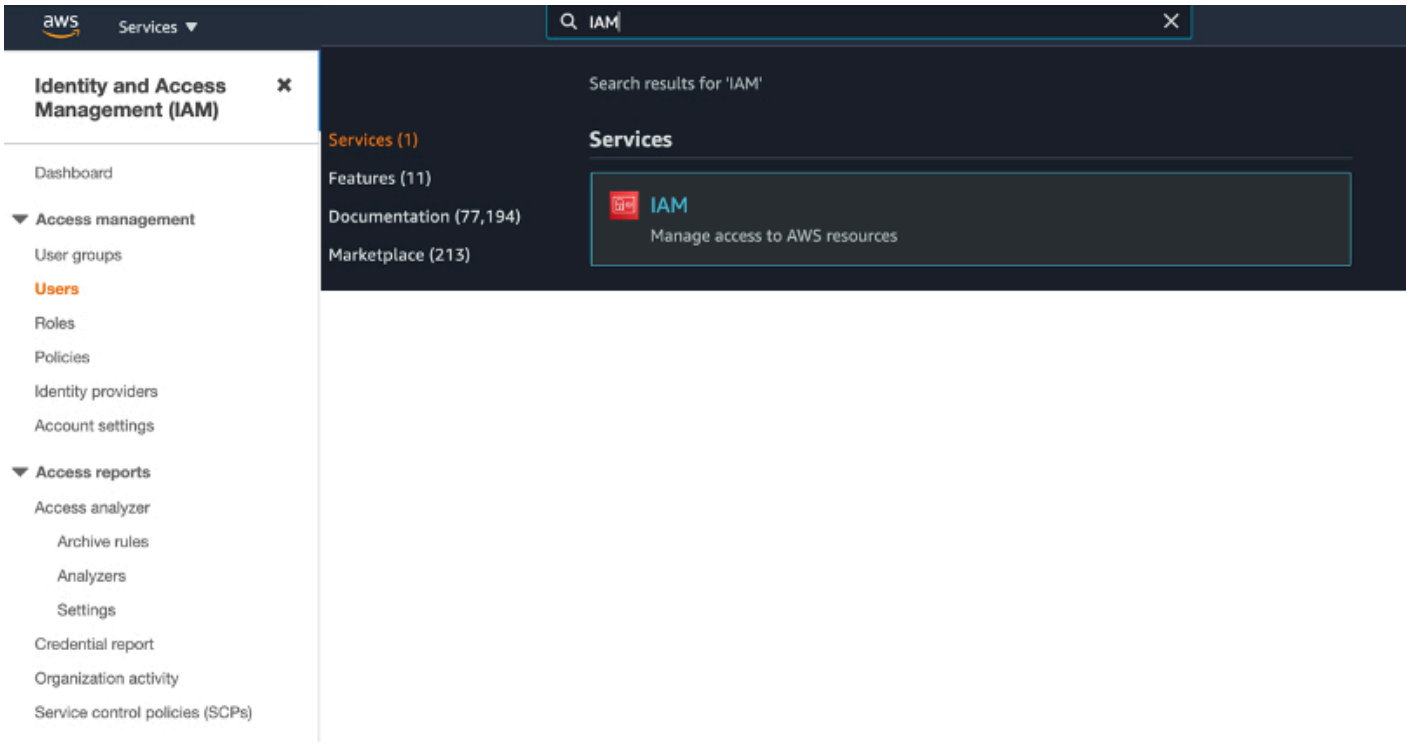
For additional information, see:

- AWS Architecture Center
- IAM documentation
- IAM policies for Amazon EC2
- Granting IAM users required permissions for Amazon EC2 resources
- Amazon Resource Names (ARNs)
- Organizing Your AWS Environment Using Multiple Accounts

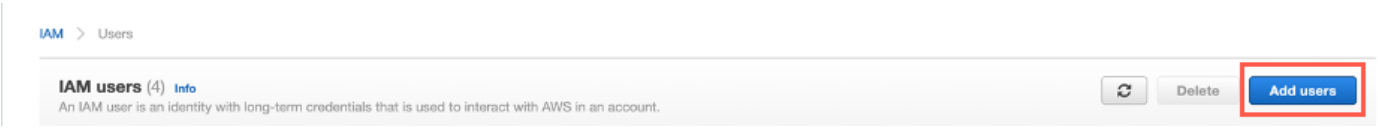
Appendix A: Adding IAM user policies

This section describes how to add IAM user policies to an AWS account. For more information, see Creating an IAM user in your AWS account in the *User Guide*.

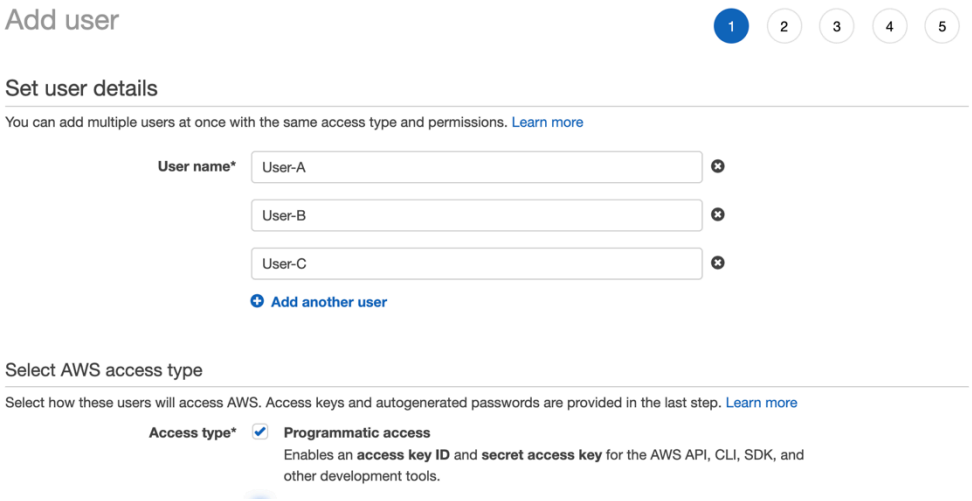
1. In the AWS Management Console, choose **Services > IAM**.



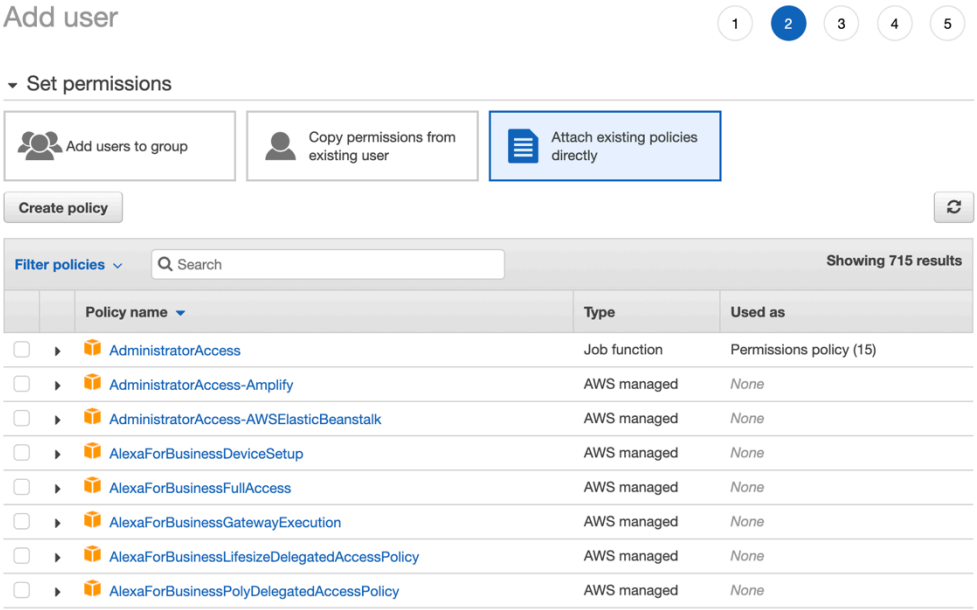
2. Choose **Users**.
3. Choose **Add Users**.



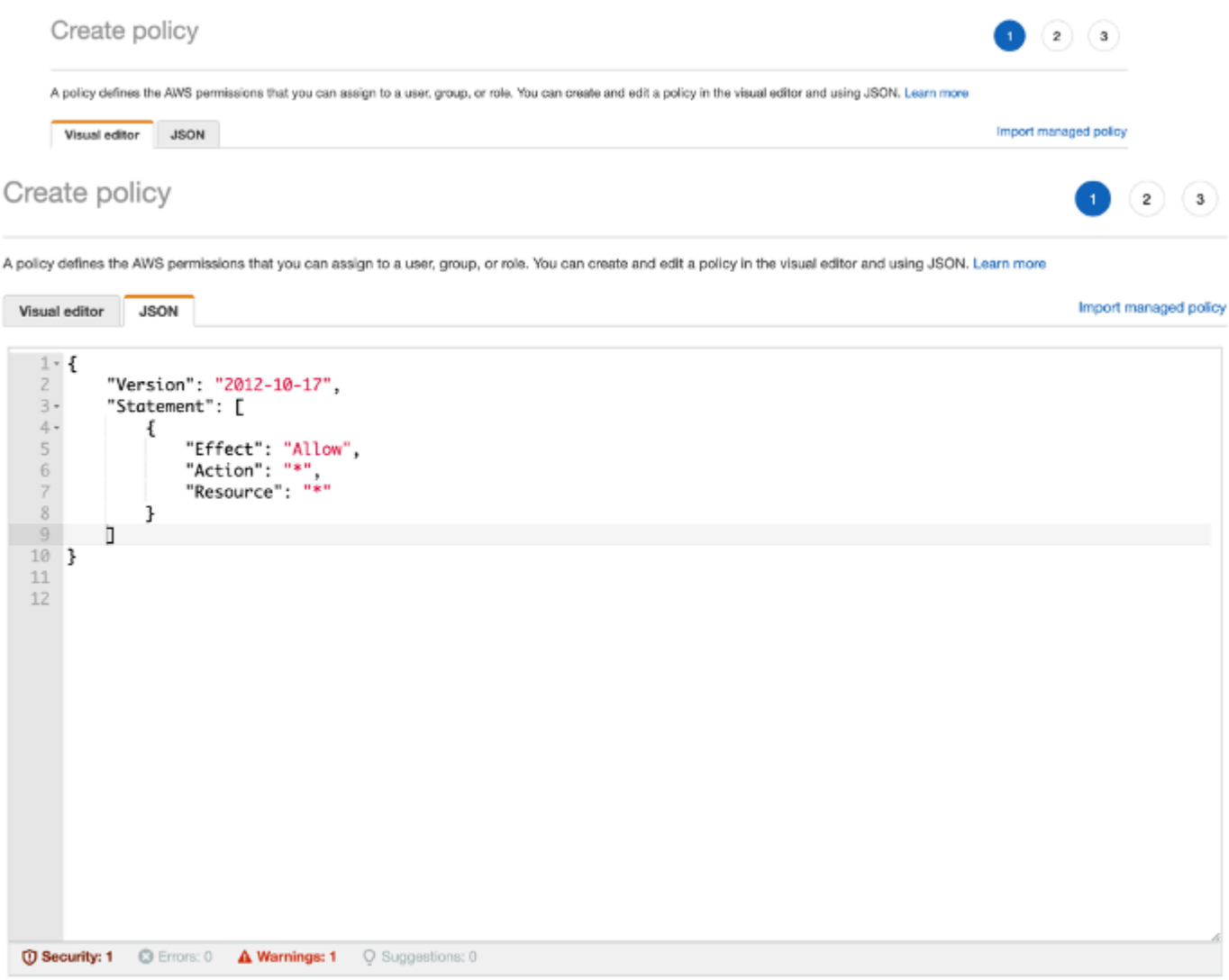
4. Enter name of the IAM user to be created.
5. Choose **Next**.



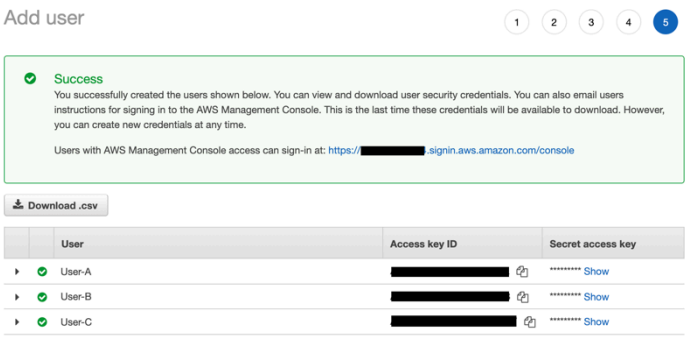
6. Choose **Next** to attach a user policy.



7. If none of these policies work for your use case, you can Create a policy and attach it. You can create the policy using the Interface, or you can create a custom JSON policy with one of the examples from Appendix B.



8. Paste the proper policy from Appendix B.
9. Choose **Apply Policy**.
10. On the previous screen, refresh the policy list, and attach the policy you just created.
11. Choose **Download Credentials**. Save the downloaded file in a secure location, as these are the user's access key ID and secret access key. They will need these to use the AWS API.


Add user
1 2 3 4 5
Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.
Users with AWS Management Console access can sign-in at: https://[redacted].signin.aws.amazon.com/console
Download .csv
User
Access key ID
Secret access key
User-A
[redacted]
********** Show
User-B
[redacted]
********** Show
User-C
[redacted]
********** Show

Appendix B: Example IAM user policies

This section provides example IAM user policies for a class that uses AWS services, including policies for the professor, teaching assistant, and students. These policies are useful for setting up the “Limited User Access to AWS Management Console” and “Separate AWS Account for Each User” scenarios described earlier in this whitepaper. For more information about policies, see Policies and permissions in IAM.

Example policies for professor (administrator)

- Full administrator access:

```
{
  "Statement": [
  {
    "Effect": "Allow",
    "Action": "*",
    "Resource": "*"
  }]
}
```

- Billing access:

```
{
  "Statement": [
  {
    "Effect": "Allow",
     "Action": [
      "aws-portal:ViewBilling"
    ],
    "Resource": "*"
  }]
}
```

- Usage access (Example Policies for Teaching Assistant):

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

```
    "Action": [
        "aws-portal:ViewUsage"
    ],
    "Resource": "*"
  }]
}
```

- Full administrator access but no access for billing or usage information:

```
{
  "Statement":[{
    "Effect":"Allow",
    "Action":"*",
    "Resource":"*"
  },
  {
    "Effect":"Deny",
    "Action":"aws-portal:*", "Resource":"*"
  }]
}
```

Example Policies for Students

- Permission to create and describe Amazon EBS volumes:

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateVolume",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
```

```
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
     }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:volume/*"
   }
   ]
}
```

- Permission to create and modify Amazon EC2 instances:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateKeyPair"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
```

```
            "Resource": "*"
        }
    ]
}
```

- Prevents modifying resource tags:

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Action": [
       "ec2:CreateTags",
        "ec2:DeleteTags"
    ],
    "Resource": [ "*"],
  "Effect": "Deny"
  }]
}
```

- For instances with a student tag, allows students to restart, stop, reboot, attach volumes, and detach volumes. If the professor or teaching assistant applies a student tag with the value being the IAM user name of specific students to specific instances, then those students can stop, reboot, attach volumes to, and detach volumes to those instances. They can also start instances that they stopped (that still have the student tag on them), but they can't start new ones.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Action": [
      "ec2:StartInstances",
      "ec2:StopInstances",
      "ec2:RebootInstances",
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
  "Condition": {
    "StringEquals": {
      "ec2:ResourceTag/Student":"${aws:username}"
    }
  },
  "Resource": [
```

```
    "arn:aws:ec2:region:account:instance/*",
    "arn:aws:ec2:region:account:volume/*"
  ],
  "Effect": "Allow"
  }]
}
```

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper updated	Minor editorial changes made throughout.	July 13, 2023
Whitepaper updated	Updated for technical accuracy.	September 15, 2021
Initial publication	Whitepaper first published.	October 1, 2013

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

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

AWS Glossary

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