Exploring AWS Essentials:

A Hands-On Journey into Cloud Computing

27 November 2024

Taeko Harada

Table of Contents

Project objectives

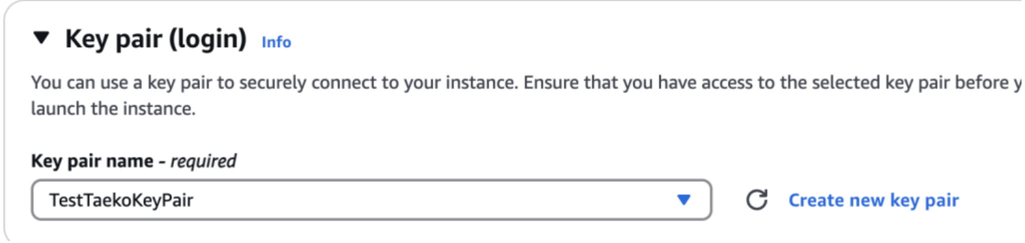
Although I hold an AWS certification, I have not yet had the opportunity to work on practical, hands-on projects. To enhance my practical skills for real-world application, I decided to explore key AWS features through hands-on practice. My study for AWS certification has provided a solid foundation to understand and apply these skills effectively.

EC2

Objective: Deepen my understanding of virtual servers.

Create a EC2 Instance

Create a key pair to connect by OpenSSH



▼ Key pair (login) Info

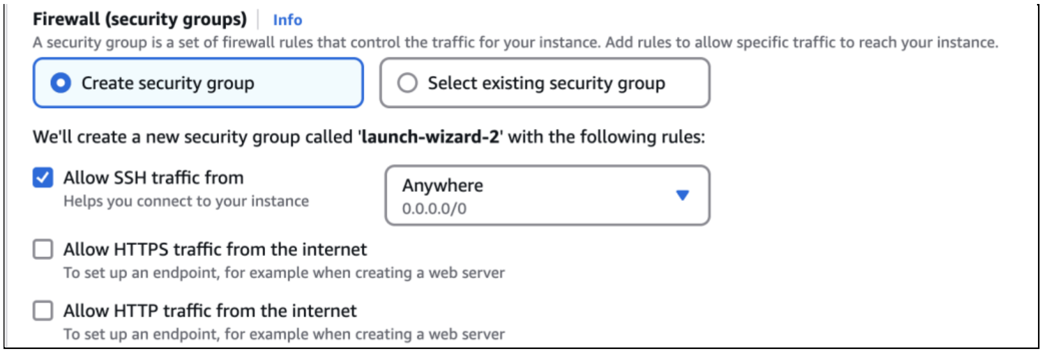
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before y launch the instance.

Key pair name - *required*

TestTaekoKeyPair ▼

Security group setting

The connection is SSH which does not allow HTTP or HTTPS for hosting web sites.



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

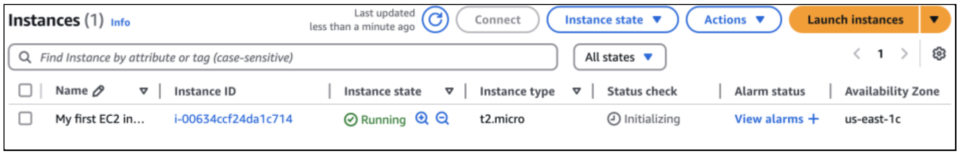
We'll create a new security group called '**launch-wizard-2**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0 ▼
Helps you connect to your instance

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

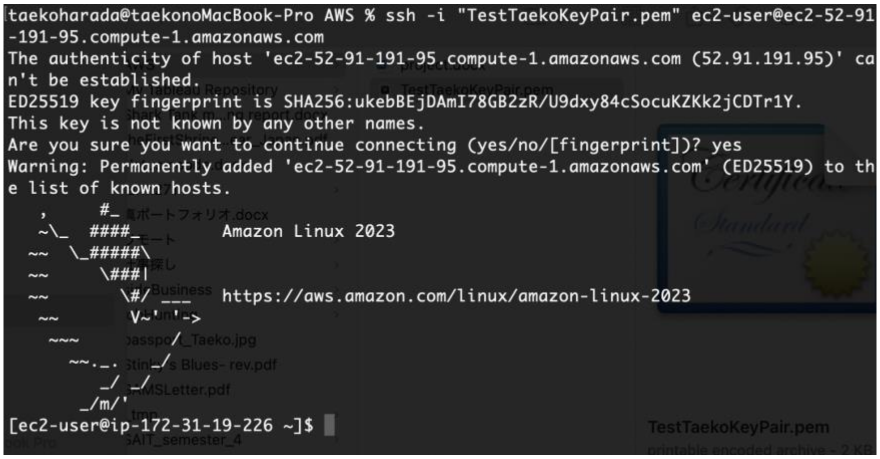
Instance list



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	My first EC2 in...	i-00634ccf24da1c714	Running	t2.micro	Initializing	View alarms +	us-east-1c

Connect to EC2 Instance

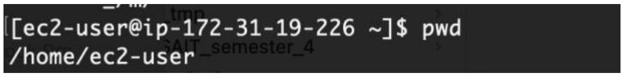
Use SSH to connect



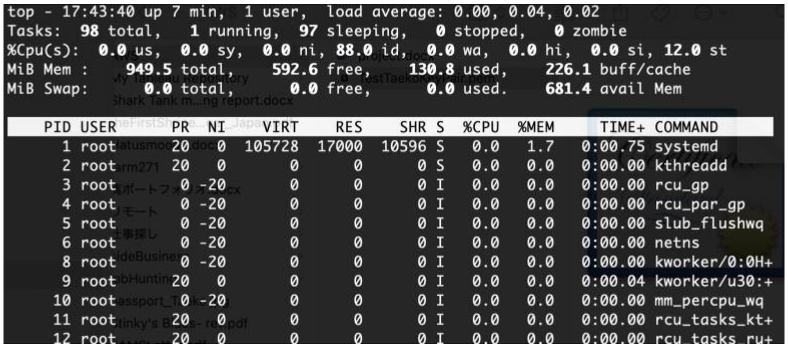
```
taekoharada@taekonoMacBook-Pro AWS % ssh -i "TestTaekoKeyPair.pem" ec2-user@ec2-52-91
-191-95.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-191-95.compute-1.amazonaws.com (52.91.191.95)' ca
n't be established.
ED25519 key fingerprint is SHA256:ukebBEjDAmI78GB2zR/U9dxy84cSocuKZKk2jCDTr1Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-191-95.compute-1.amazonaws.com' (ED25519) to th
e list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-19-226 ~]$ 
```

Basic Commands

Explore the instance with commands like ls, pwd, and top



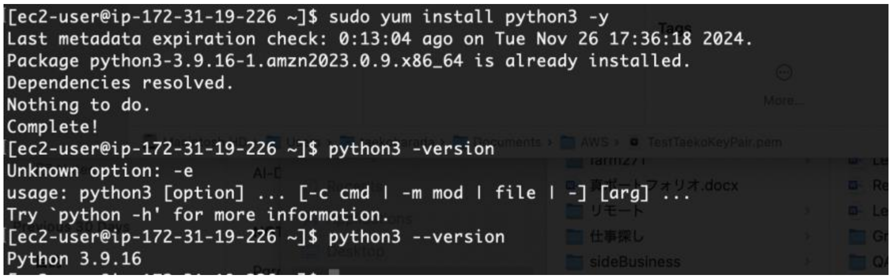
```
[ec2-user@ip-172-31-19-226 ~]$ pwd
/home/ec2-user
```



```
top - 17:43:40 up 7 min,  1 user,  load average: 0.00, 0.04, 0.02
Tasks:  98 total,   1 running,  97 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 88.0 id,  0.0 wa,  0.0 hi,  0.0 si, 12.0 st
MiB Mem :    949.5 total,    592.6 free,    130.8 used,    226.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    681.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  105728  17000  10596 S   0.0   1.7   0:00.75 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H+
      9 root      20   0       0      0      0 I   0.0   0.0   0:00.04 kworker/u30:+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kt+
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_ru+
```

Install software on EC2 instance

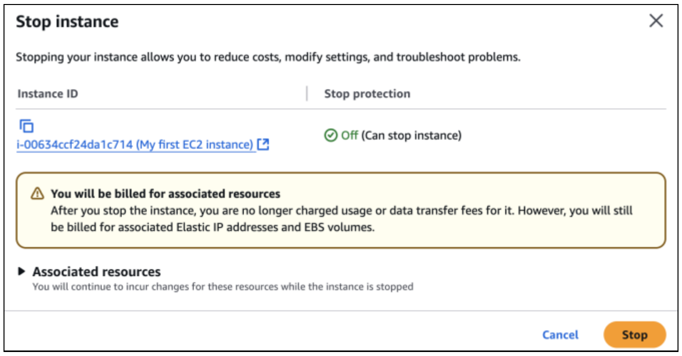
Install python



```
[ec2-user@ip-172-31-19-226 ~]$ sudo yum install python3 -y
Last metadata expiration check: 0:13:04 ago on Tue Nov 26 17:36:18 2024.
Package python3-3.9.16-1.amzn2023.0.9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-19-226 ~]$ python3 -version
Unknown option: -e
usage: python3 [option] ... [-c cmd | -m mod | file | -] [arg] ...
Try `python -h' for more information.
[ec2-user@ip-172-31-19-226 ~]$ python3 --version
Python 3.9.16
```

Stop/Restart the Instance

Stopping the EC2 instance from the AWS Management Console



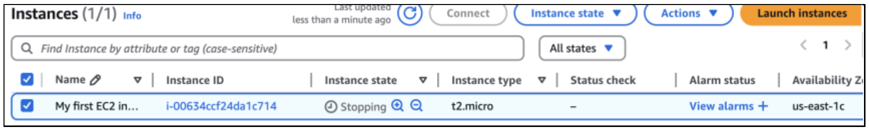
Stop instance

Stopping your instance allows you to reduce costs, modify settings, and troubleshoot problems.

Instance ID	Stop protection
i-00634ccf24da1c714 (My first EC2 instance)	Off (Can stop instance)

You will be billed for associated resources
After you stop the instance, you are no longer charged usage or data transfer fees for it. However, you will still be billed for associated Elastic IP addresses and EBS volumes.

Associated resources
You will continue to incur changes for these resources while the instance is stopped

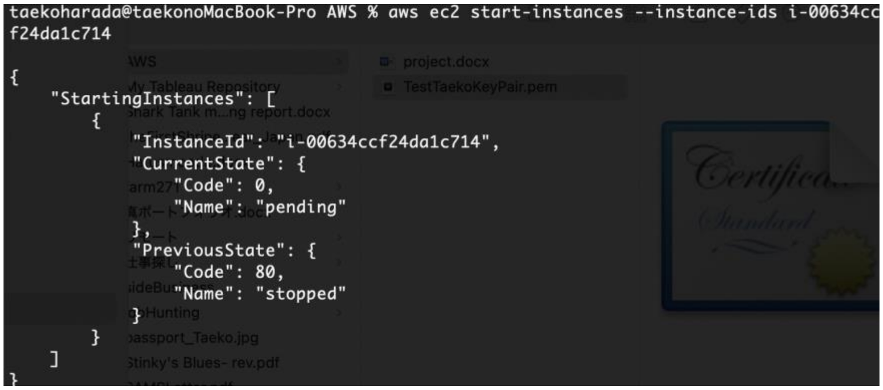


Instances (1/1)

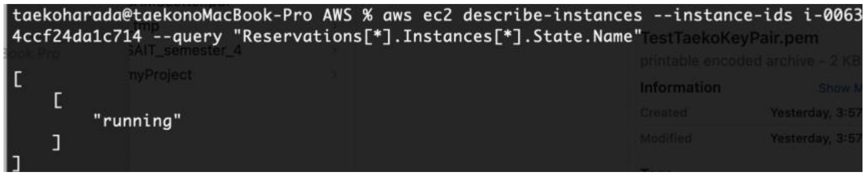
Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
☑	My first EC2 in...	i-00634ccf24da1c714	Stopping	t2.micro	–	View alarms +	us-east-1c

Restarting the EC2 instance from CLI



```
taekoharada@taekonoMacBook-Pro AWS % aws ec2 start-instances --instance-ids i-00634cc
f24da1c714

{
    "StartingInstances": [
        {
            "InstanceId": "i-00634ccf24da1c714",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```



```
taekoharada@taekonoMacBook-Pro AWS % aws ec2 describe-instances --instance-ids i-0063
4ccf24da1c714 --query "Reservations[*].Instances[*].State.Name"
[
    [
        "running"
    ]
]
```

S3 (Simple Storage Service)

Objective: Learn how to store and retrieve files.

Create a Bucket

Object Ownership

ACL (Access Control List) is set to 'disable'. ACL is for granting permission (read/write).

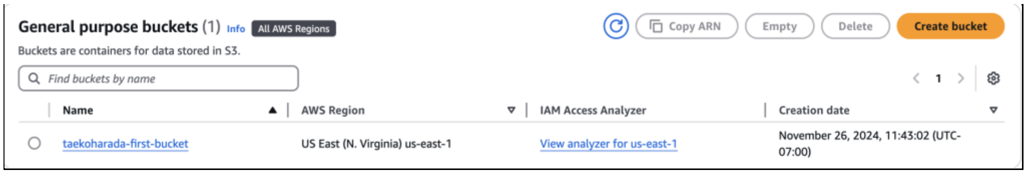
Block Public Access settings for this bucket

Set to 'Block all public accesses.

Bucket Versioning

Versioning is for restore. Set to 'Disable'.

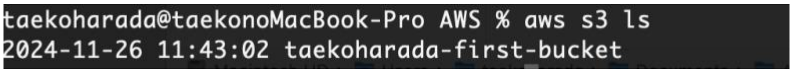
The list of Buckets



General purpose buckets (1)

Buckets are containers for data stored in S3.

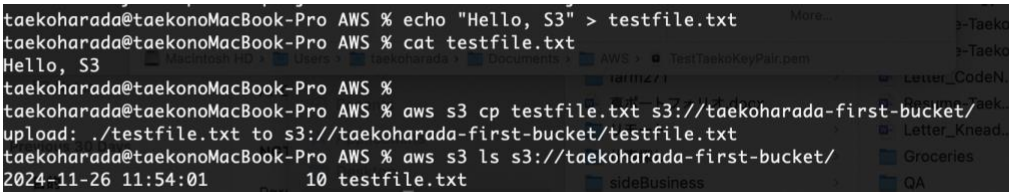
	Name	AWS Region	IAM Access Analyzer	Creation date
○	taekoharada-first-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	November 26, 2024, 11:43:02 (UTC-07:00)



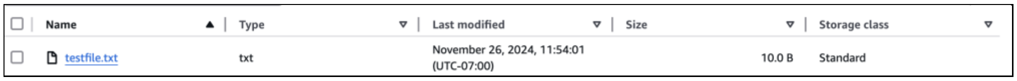
```
taekoharada@taekonoMacBook-Pro AWS % aws s3 ls
2024-11-26 11:43:02 taekoharada-first-bucket
```

Upload Files

Create a text file and upload to S3.



```
taekoharada@taekonoMacBook-Pro AWS % echo "Hello, S3" > testfile.txt
taekoharada@taekonoMacBook-Pro AWS % cat testfile.txt
Hello, S3
taekoharada@taekonoMacBook-Pro AWS %
taekoharada@taekonoMacBook-Pro AWS % aws s3 cp testfile.txt s3://taekoharada-first-bucket/
upload: ./testfile.txt to s3://taekoharada-first-bucket/testfile.txt
taekoharada@taekonoMacBook-Pro AWS % aws s3 ls s3://taekoharada-first-bucket/
2024-11-26 11:54:01         10 testfile.txt
```

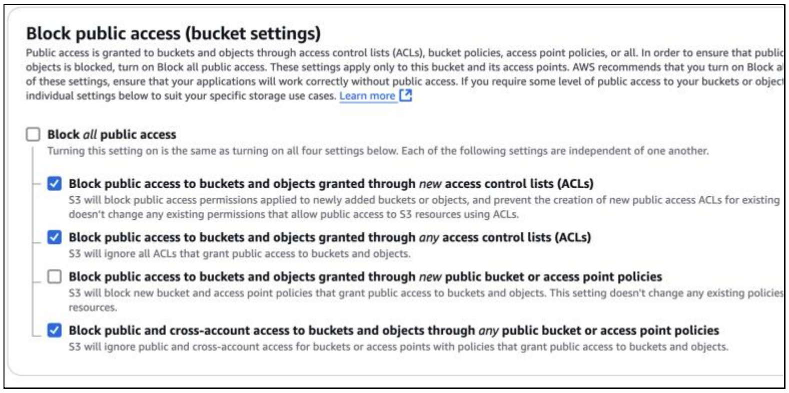


□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	testfile.txt	txt	November 26, 2024, 11:54:01 (UTC-07:00)	10.0 B	Standard

Access the File

Make the file public and use the URL to view/download it.

To add a new bucket policy allows the file access from public, it is necessary to change Block Public Access setting.



Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block al of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or object individual settings below to suit your specific storage use cases. Learn more

□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

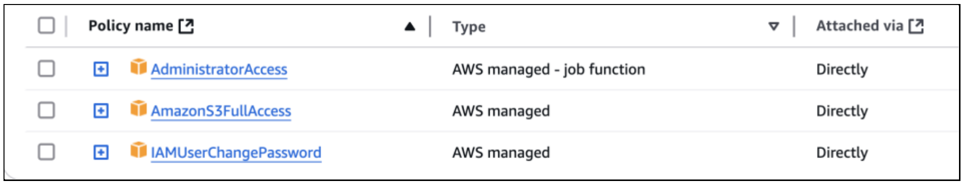
☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.

□ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies resources.

☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

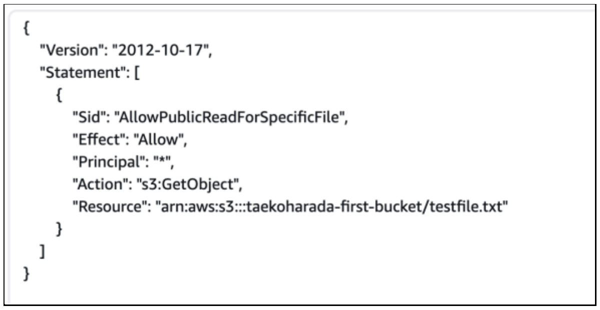
Grant the necessary permissions to the user responsible for setting the bucket policy.
Assign the **AmazonS3FullAccess** policy to the user.
This policy includes the permissions **s3:PutBucketPolicy** and **s3:GetBucketPolicy**.



□		Policy name ▲	Type ▽	Attached via
□	⊞	AdministratorAccess	AWS managed - job function	Directly
□	⊞	AmazonS3FullAccess	AWS managed	Directly
□	⊞	IAMUserChangePassword	AWS managed	Directly

Set Bucket Policy

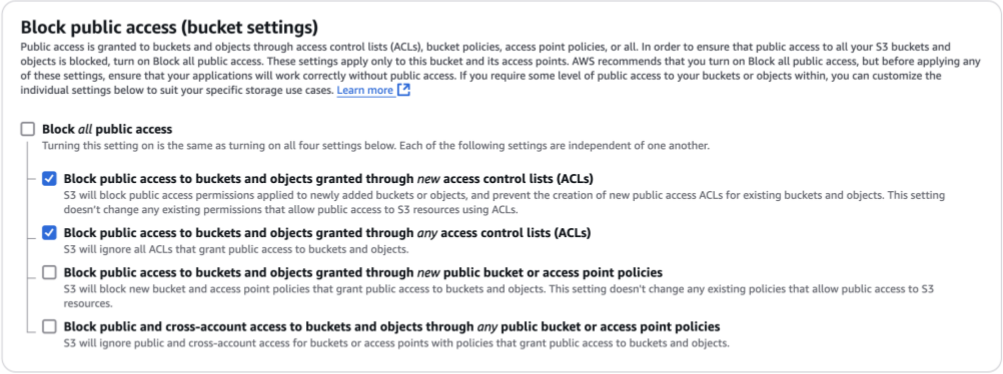
Finally, admin-user can edit bucket policy.



```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowPublicReadForSpecificFile",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::taekoharada-first-bucket/testfile.txt"
      }
   ]
}
```

Allow public access to testfile.txt

On **Block Public Access Settings,** uncheck **"Block public and cross-account access to buckets and objects through *any* public bucket or access point policies"**.



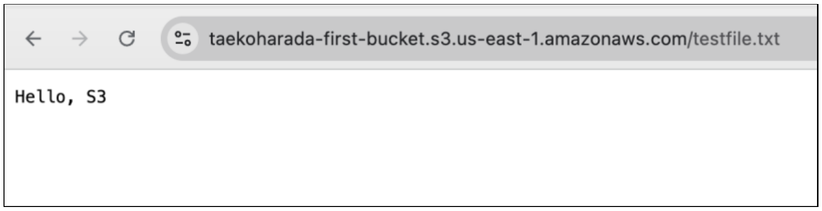
Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Access to testfile.txt in the browser.



taekoharada-first-bucket.s3.us-east-1.amazonaws.com/testfile.txt

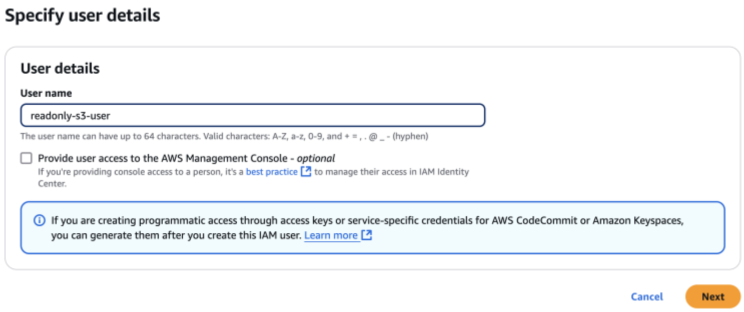
```
Hello, S3
```

IAM (Identity and Access Management)

Objective: Manage user permissions and policies.

Create a new IAM user with read-only permission to S3

Create a new user named ‘readonly-s3-user’



Specify user details

User details

User name

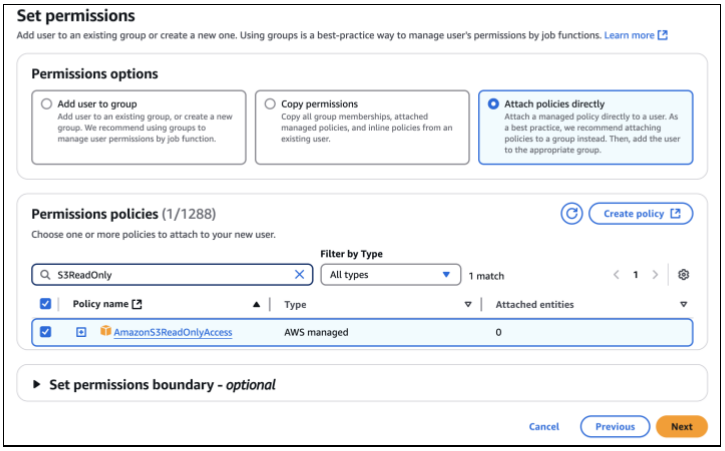
readonly-s3-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

Set permission, ‘AmazonS3ReadOnlyAccess’.



Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1288)

Choose one or more policies to attach to your new user.

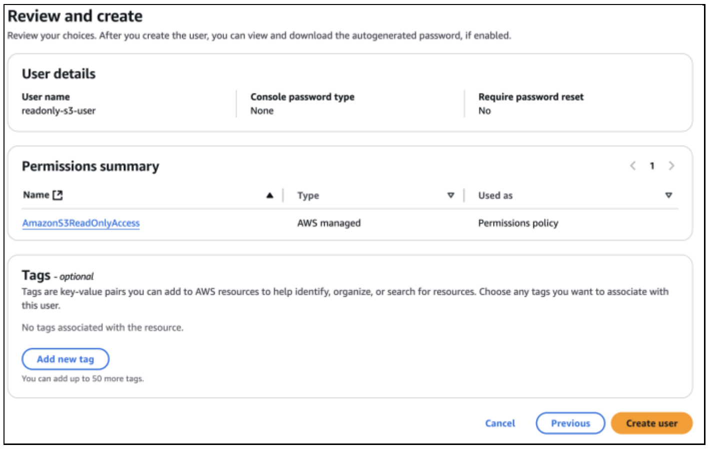
S3ReadOnly

Filter by Type: All types

1 match

☑	Policy name	Type	Attached entities
☑	AmazonS3ReadOnlyAccess	AWS managed	0

▶ **Set permissions boundary** - *optional*



Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
readonly-s3-user	None	No

Permissions summary

Name	Type	Used as
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy

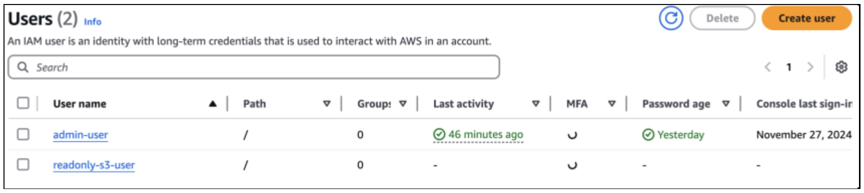
Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.

Existing user list

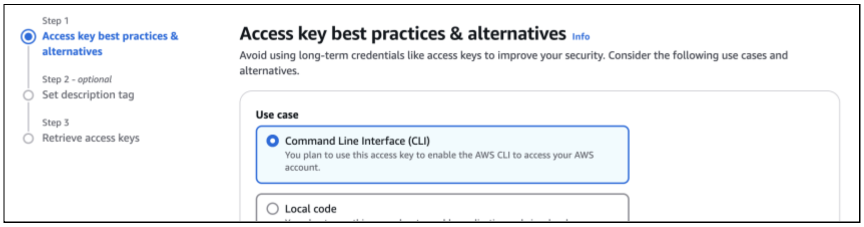


Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name ▲	Path ▽	Group: ▽	Last activity ▽	MFA ▽	Password age ▽	Console last sign-ir
□	admin-user	/	0	✓ 46 minutes ago	◡	✓ Yesterday	November 27, 2024
□	readonly-s3-user	/	0	-	◡	-	-

Create Access Key



Step 1
● Access key best practices & alternatives

Step 2 - *optional*
○ Set description tag

Step 3
○ Retrieve access keys

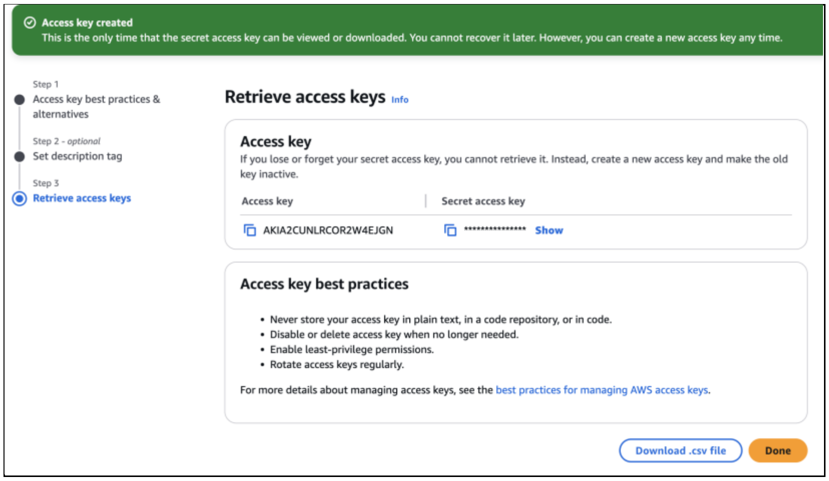
Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code



✓ **Access key created**
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
● Access key best practices & alternatives

Step 2 - *optional*
● Set description tag

Step 3
● Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

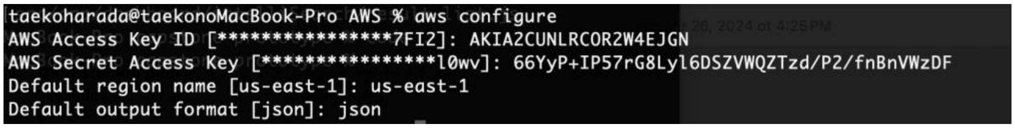
Access key	Secret access key
AKIA2CUNLRCOR2W4EJGN	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

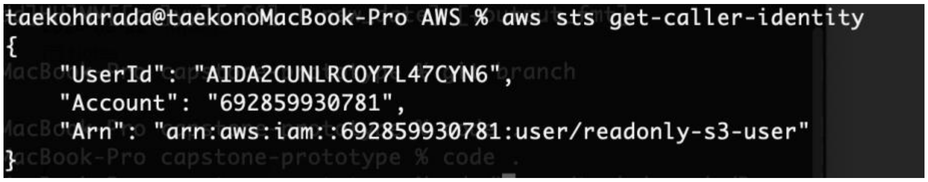
For more details about managing access keys, see the best practices for managing AWS access keys.

Configure AWS CLI for the User



```
taekoharada@taekonoMacBook-Pro AWS % aws configure
AWS Access Key ID [****************7FI2]: AKIA2CUNLRCOR2W4EJGN
AWS Secret Access Key [****************l0wv]: 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
Default region name [us-east-1]: us-east-1
Default output format [json]: json
```

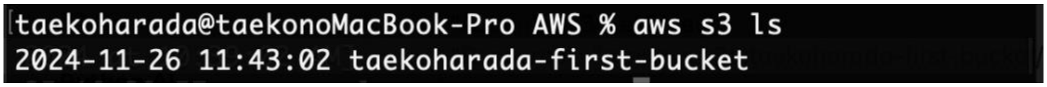
Verify which user I am using.



```
taekoharada@taekonoMacBook-Pro AWS % aws sts get-caller-identity
{
    "UserId": "AIDA2CUNLRCOY7L47CYN6",
    "Account": "692859930781",
    "Arn": "arn:aws:iam::692859930781:user/readonly-s3-user"
}
```

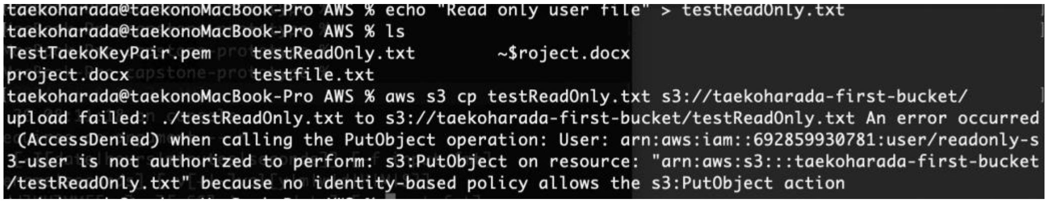
Test the Read-Only User

The user should see a list of S3 buckets.



```
taekoharada@taekonoMacBook-Pro AWS % aws s3 ls
2024-11-26 11:43:02 taekoharada-first-bucket
```

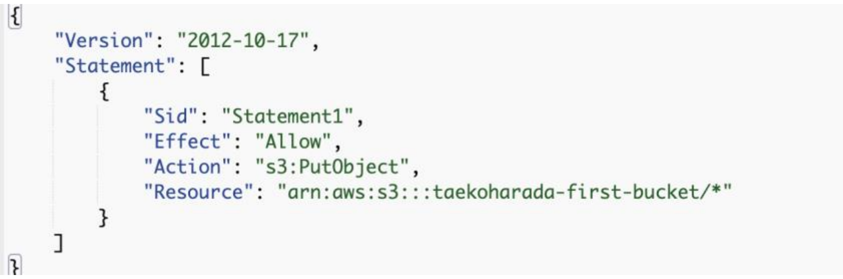
Uploading a file to S3 bucket failed.



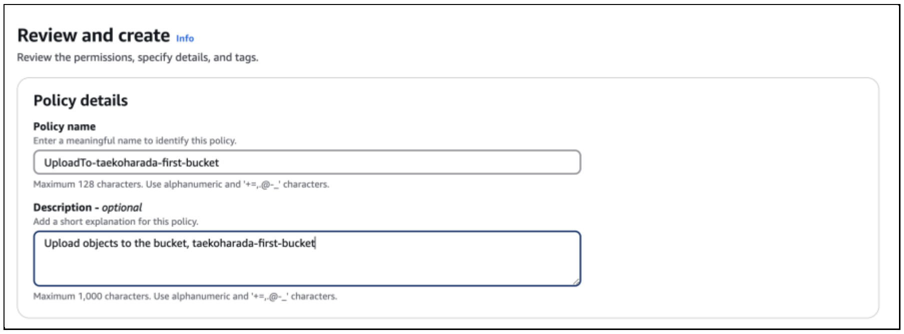
```
taekoharada@taekonoMacBook-Pro AWS % echo "Read only user file" > testReadOnly.txt
taekoharada@taekonoMacBook-Pro AWS % ls
TestTaekoKeyPair.pem    testReadOnly.txt        ~$roject.docx
project.docx            testfile.txt
taekoharada@taekonoMacBook-Pro AWS % aws s3 cp testReadOnly.txt s3://taekoharada-first-bucket/
upload failed: ./testReadOnly.txt to s3://taekoharada-first-bucket/testReadOnly.txt An error occurred
 (AccessDenied) when calling the PutObject operation: User: arn:aws:iam::692859930781:user/readonly-s
3-user is not authorized to perform: s3:PutObject on resource: "arn:aws:s3:::taekoharada-first-bucket
/testReadOnly.txt" because no identity-based policy allows the s3:PutObject action
```

Use IAM Policies

Create a **custom** policy to grant the permission, ‘s3:PutObject’.



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::taekoharada-first-bucket/*"
        }
    ]
}
```



Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

UploadTo-taekoharada-first-bucket

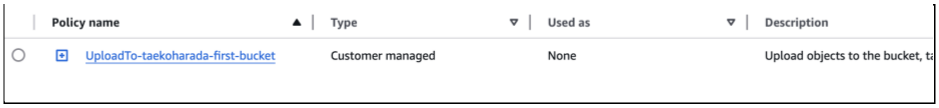
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - *optional*

Add a short explanation for this policy.

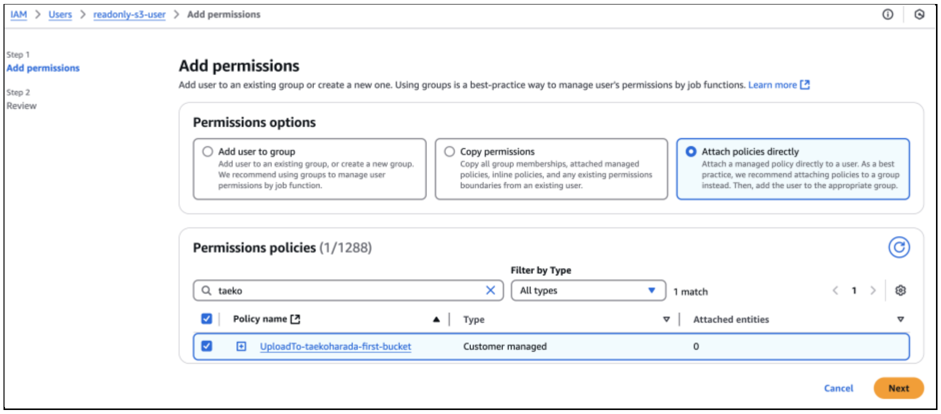
Upload objects to the bucket, taekoharada-first-bucket

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.



	Policy name ▲	Type ▽	Used as ▽	Description
○	UploadTo-taekoharada-first-bucket	Customer managed	None	Upload objects to the bucket, ta

Attach the custom policy to ‘readonly-s3-user’



IAM > Users > readonly-s3-user > Add permissions

Step 1
Add permissions

Step 2
Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

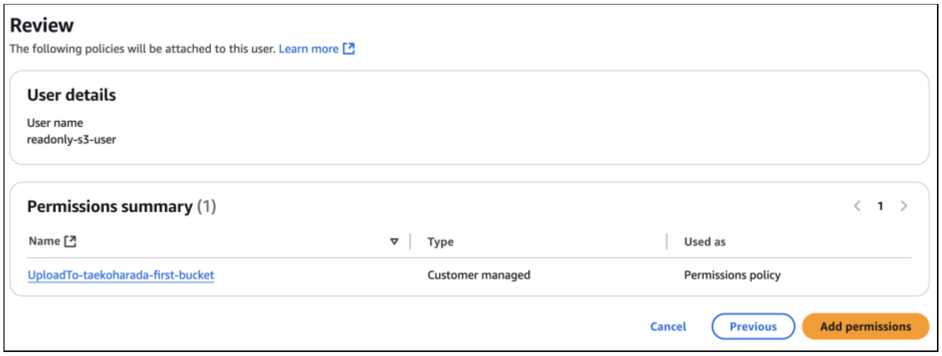
Permissions policies (1/1288)

Search: taeko

Filter by Type: All types

1 match

☑	Policy name	Type	Attached entities
☑	UploadTo-taekoharada-first-bucket	Customer managed	0



Review

The following policies will be attached to this user. Learn more

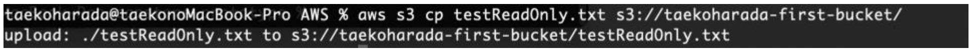
User details

User name
readonly-s3-user

Permissions summary (1)

Name	Type	Used as
UploadTo-taekoharada-first-bucket	Customer managed	Permissions policy

Upload the file to the bucket again. The file was successfully uploaded.



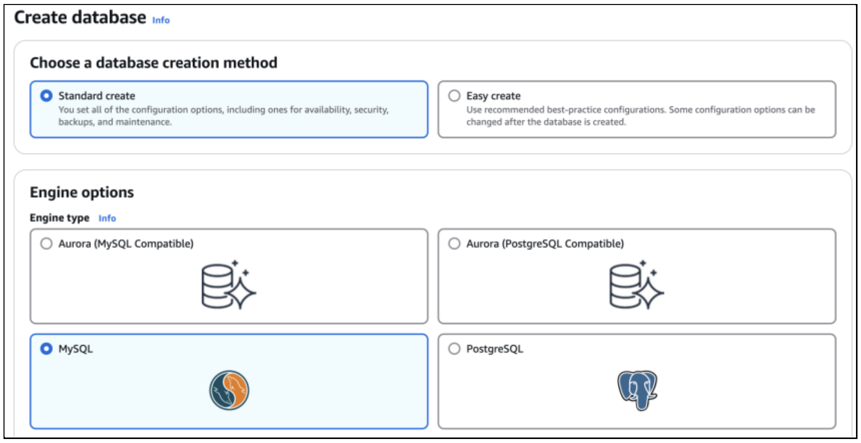
```
taekoharada@taekonoMacBook-Pro AWS % aws s3 cp testReadOnly.txt s3://taekoharada-first-bucket/
upload: ./testReadOnly.txt to s3://taekoharada-first-bucket/testReadOnly.txt
```

RDS (Relational Database Service)

Objective: Set up a managed database.

Launch an RDS Instance

Choose a database engine, MySQL.



Create database Info

Choose a database creation method

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

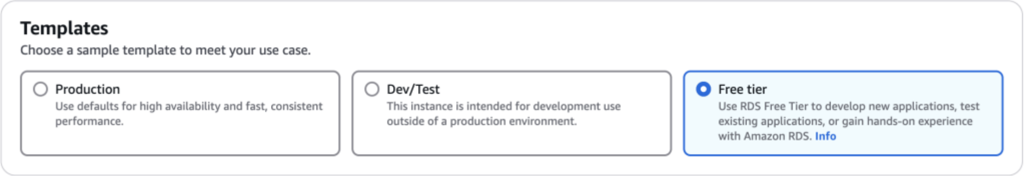
○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ PostgreSQL

Choose Free tier.



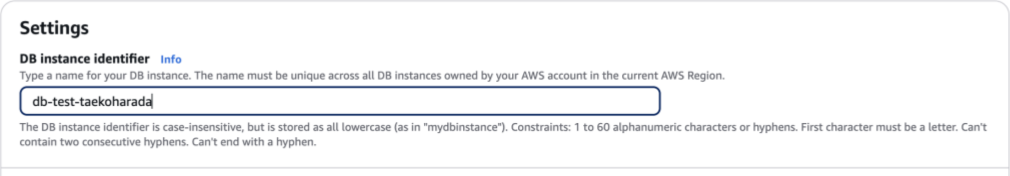
Templates
Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

Name the database instance.



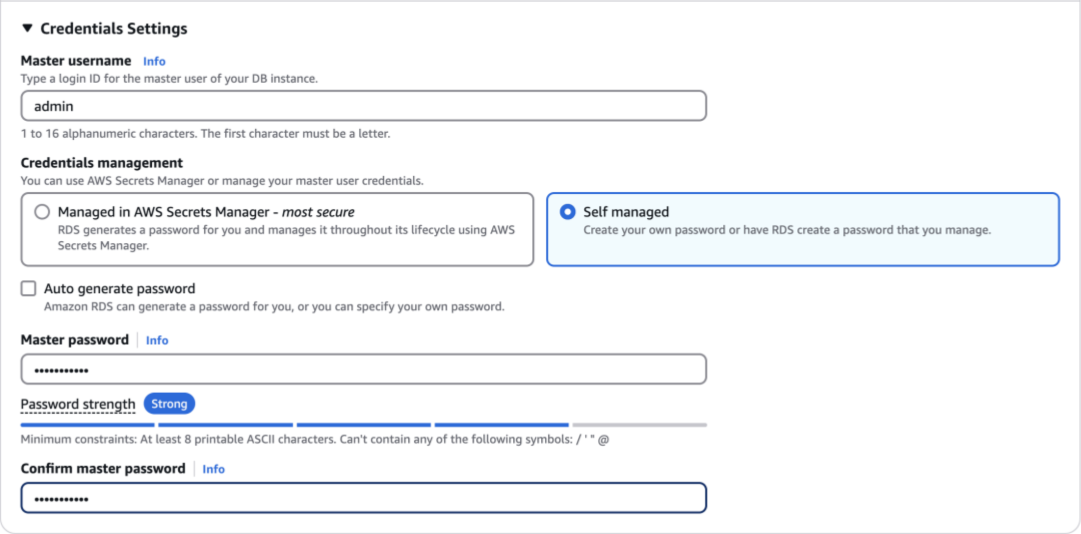
Settings

DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

db-test-taekoharada

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Set database user name and password.



▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password | Info

•••••••••••

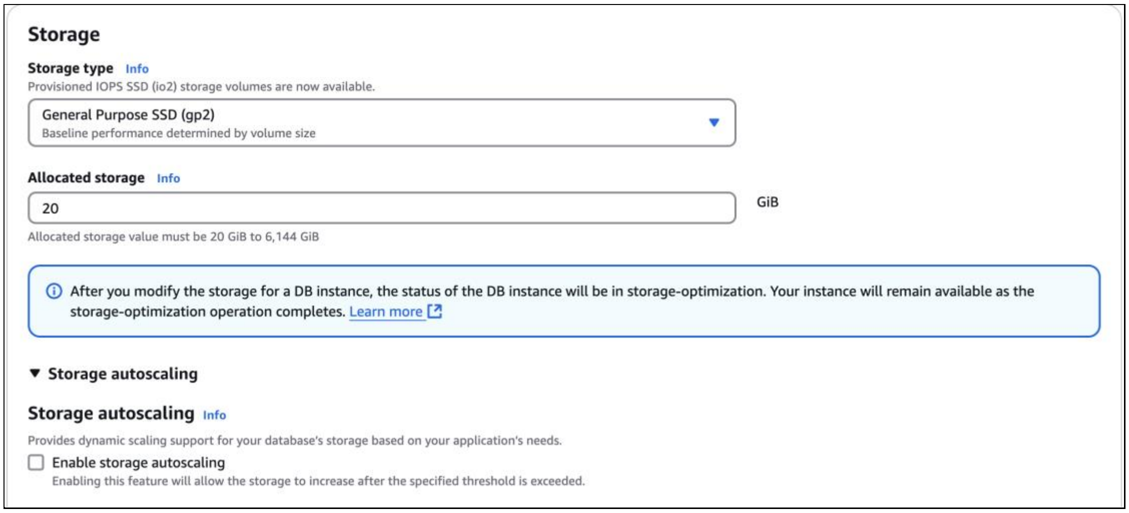
Password strength Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password | Info

•••••••••••

Uncheck ‘Enable storage autoscaling’.



Storage

Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

20 GiB

Allocated storage value must be 20 GiB to 6,144 GiB

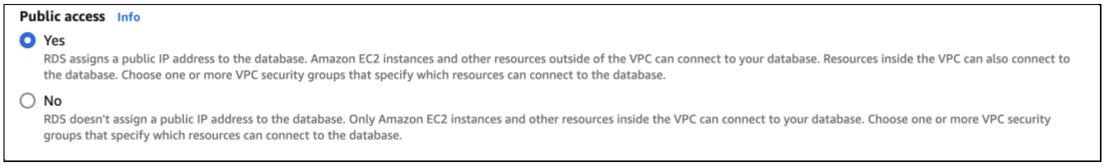
ⓘ After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. Learn more

▼ Storage autoscaling

Storage autoscaling Info
Provides dynamic scaling support for your database's storage based on your application's needs.

□ Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Allow public access for testing.

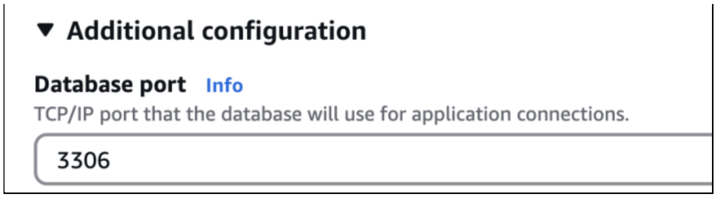


Public access Info

● Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

○ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

Verify the port number

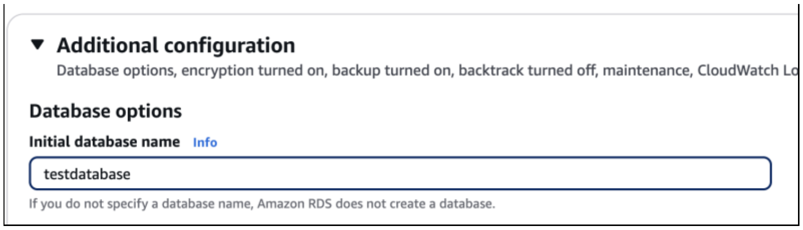


▼ Additional configuration

Database port Info
TCP/IP port that the database will use for application connections.

3306

Enter the initial database name.



▼ Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Lo

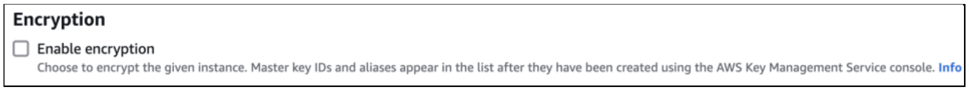
Database options

Initial database name Info

testdatabase

If you do not specify a database name, Amazon RDS does not create a database.

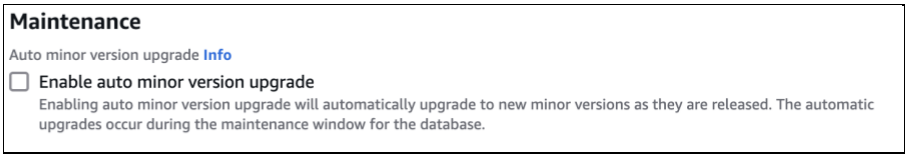
For free tier setting, uncheck these options.



Encryption

□ Enable encryption

Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info



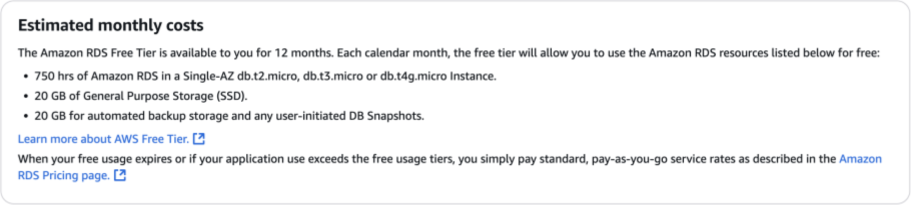
Maintenance

Auto minor version upgrade Info

□ Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Verify the estimated cost.



Estimated monthly costs

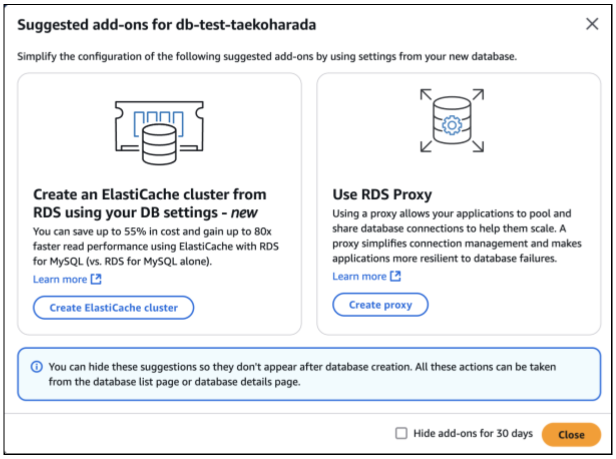
The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.

Add-ons (Not Free Tier Eligible)



Suggested add-ons for db-test-taekoharada

Simplify the configuration of the following suggested add-ons by using settings from your new database.

Create an ElastiCache cluster from RDS using your DB settings - *new*

You can save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone).

Learn more

Use RDS Proxy

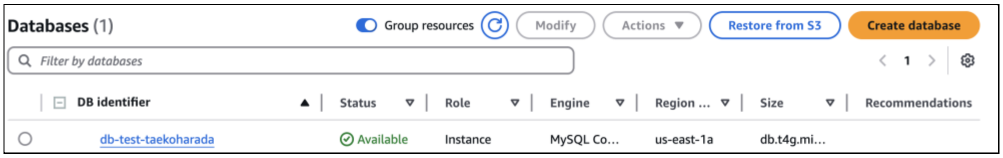
Using a proxy allows your applications to pool and share database connections to help them scale. A proxy simplifies connection management and makes applications more resilient to database failures.

Learn more

You can hide these suggestions so they don't appear after database creation. All these actions can be taken from the database list page or database details page.

ElastiCache cluster: Caching service to improve database performance.

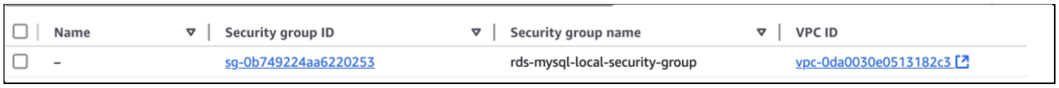
RDS Proxy: The proxy creates a pool of connections to reduces the overhead on the database and improves response times.



	DB identifier ▲	Status	Role	Engine	Region ...	Size	Recommendations
○	db-test-taekoharada	✓ Available	Instance	MySQL Co...	us-east-1a	db.t4g.mi...	

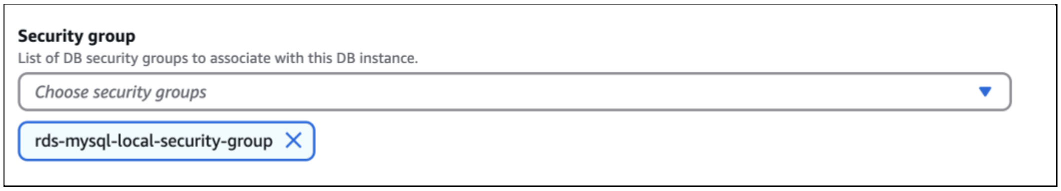
Connect to the Database

For connecting from local, create a new security group named ‘rds-mysql-local-security-group’.



□	Name	Security group ID	Security group name	VPC ID
□	–	sg-0b749224aa6220253	rds-mysql-local-security-group	vpc-0da0030e0513182c3

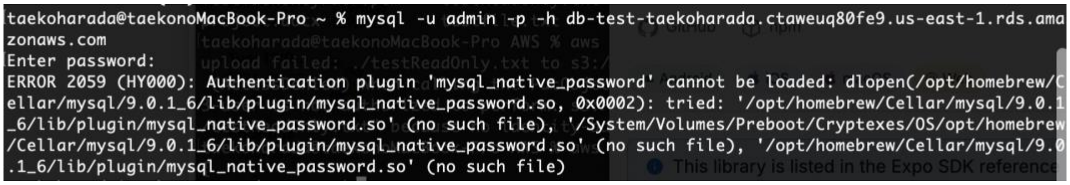
Modify the database’s security group.



Security group
List of DB security groups to associate with this DB instance.

rds-mysql-local-security-group ✕

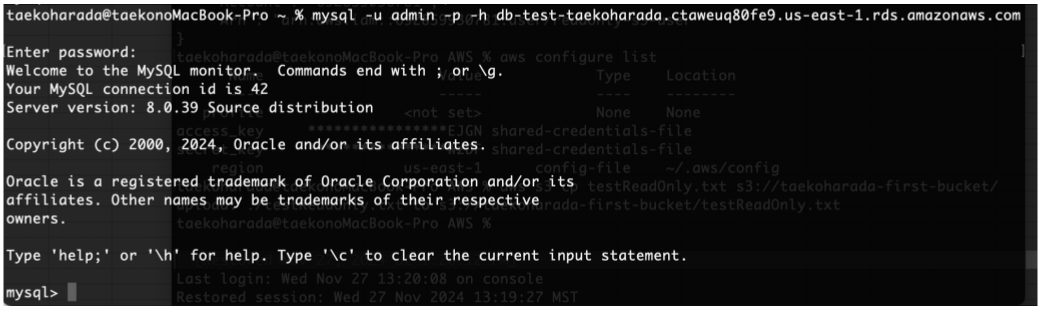
Fail to the connection.



```
taekoharada@taekonoMacBook-Pro ~ % mysql -u admin -p -h db-test-taekoharada.ctaweuq80fe9.us-east-1.rds.ama
zonaws.com
Enter password:
ERROR 2059 (HY000): Authentication plugin 'mysql_native_password' cannot be loaded: dlopen(/opt/homebrew/C
ellar/mysql/9.0.1_6/lib/plugin/mysql_native_password.so, 0x0002): tried: '/opt/homebrew/Cellar/mysql/9.0.1
_6/lib/plugin/mysql_native_password.so' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/opt/homebrew
/Cellar/mysql/9.0.1_6/lib/plugin/mysql_native_password.so' (no such file), '/opt/homebrew/Cellar/mysql/9.0
.1_6/lib/plugin/mysql_native_password.so' (no such file)
```

Install mysql@8.0 to use the authentication, ‘mysql_native_password’.

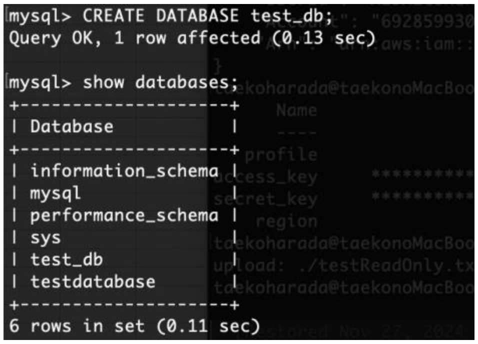
Successfully, connected to RDS from local.



```
taekoharada@taekonoMacBook-Pro ~ % mysql -u admin -p -h db-test-taekoharada.ctaweuq80fe9.us-east-1.rds.amazonaws.com

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Test Basic Queries

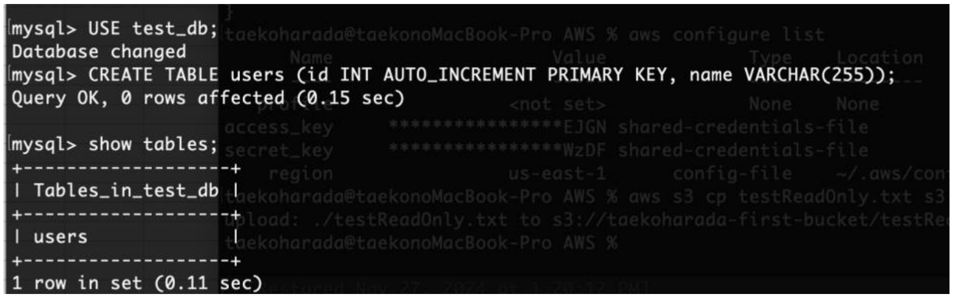
CREATE DATABASE test_db;



```
mysql> CREATE DATABASE test_db;
Query OK, 1 row affected (0.13 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test_db            |
| testdatabase       |
+--------------------+
6 rows in set (0.11 sec)
```

USE test_db;

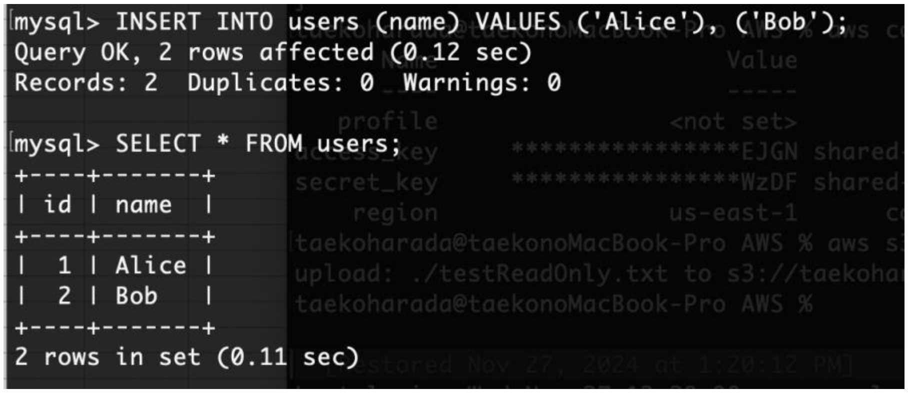
CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255));



```
mysql> USE test_db;
Database changed
mysql> CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255));
Query OK, 0 rows affected (0.15 sec)

mysql> show tables;
+-------------------+
| Tables_in_test_db |
+-------------------+
| users             |
+-------------------+
1 row in set (0.11 sec)
```

INSERT INTO users (name) VALUES ('Alice'), ('Bob');

SELECT * FROM users;



```
mysql> INSERT INTO users (name) VALUES ('Alice'), ('Bob');
Query OK, 2 rows affected (0.12 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM users;
+----+-------+
| id | name  |
+----+-------+
|  1 | Alice |
|  2 | Bob   |
+----+-------+
2 rows in set (0.11 sec)
```

Conclusion

This project allowed me to deepen my understanding of AWS through hands-on experience with main features like EC2, S3, IAM, and RDS. I learned that AWS provides robust permission configurations, which can make the settings complex and challenging. However, this complexity is necessary, as it significantly enhances security. This project provided me with the skills to confidently tackle real-world cloud computing scenarios.