

AWS S3 CUR Integration

Kubecost allows integration with your AWS CUR directly from the S3 bucket where it is saved. This feature enables users to bypass use of Athena which can have its use restricted and has a higher cost overhead than using S3 directly. The drawback of integrating directly with S3 is that there can be a reduction in performance at build time, however this is unlikely to affect users significantly outside of full Asset ETL rebuilds.

Create CUR with correct configurations

Due to limitations of the AWS Go SDK, the CUR for this configuration will need specific configurations. This means that you may not be able to use your existing CUR if you have one. Follow this [guide](#) with the specific settings detailed below.

Additional report details

When resources are created, AWS assigns each resource a unique resource ID. Including individual resource IDs in your report can significantly increase the file size.

- Include resource IDs

Data refresh settings

These charges are often related to refunds, credits, and AWS Support fees.

- Automatically refresh your Cost & Usage Report when charges are detected for previous months with closed bills.

“Include resource IDs” must be enabled, Automatic refresh is recommended but not strictly required.

Time granularity

- Hourly
- Daily
- Monthly

The time granularity on which report data are measured and displayed.

Report versioning

- Create new report version
- Overwrite existing report

Enable report data integration for

- Amazon Athena
- Amazon Redshift
- Amazon QuickSight

Compression type

GZIP ▼

File format

text/csv

Use a Daily “Time granularity”, Overwrite existing report for “Report versioning” and use GZIP “Compression type”. Integrations with Redshift and QuickSight are currently unsupported and Athena enforces parquet compress, so it is best to leave all unselected. Report name and prefix can be any value, but it is recommended that you create a new S3 bucket for this CUR and store nothing else in it.

The next few steps are very similar to the

Setting up IAM permission

Navigate to <https://console.aws.amazon.com/iam> and create a user with the following permission policy.

```
```\n{\n  "Version": "2012-10-17",\n  "Statement": [\n    {\n      "Sid": "S3CURAccess",\n      "Effect": "Allow",\n      "Action": [\n        "s3:GetObject",\n        "s3:ListBucket"\n      ],\n      "Resource": [\n
```

```
"arn:aws:s3:::<CUR-Bucket-Name>"
"arn:aws:s3:::<CUR-Bucket-Name>/*"
]
}
]
}
...
```

When creating the user, check the "Access key - Programmatic access" box so that service key credentials are created for the user. If you are attaching the policy to an existing user Access Management > Users. Find the Kubecost User and select Security Credentials > Create Access Key. Be sure to keep these values in a safe place as you will not be able to view the key again after exiting the page.

## Provide CUR configuration values to kubecost

The following values will need to be set to enable your S3 integration. While athena is used as a prefix to several of these fields, the values are being applied to the S3 SDK to access your CUR directly.

- `athenaProjectID`: e.g. "530337586277" # The AWS AccountID where the CUR is.
- `athenaBucketName`: The S3 bucket where the CUR is being saved. This value should match exactly what you entered into the bucket name field when creating the CUR with no additional prefix.
- `athenaRegion`: The aws region S3 is running in
- `serviceKeyName`: the service key name from the previous step
- `serviceKeySecret`: the service key secret from the previous step

`athenaDatabase` and `athenaTable` should be left blank as they are not relevant to the S3 configuration.

These values can be attached via the front end on the /settings.html page or via secret using the values in your helm deployment. Greater details can be found on these processes [here](#).