Example of custom WAF rules setting:

```
{
  "app": "python ./deploy/app.py",
  "context": {
    "@aws-cdk/aws-apigateway:usagePlanKeyOrderInsensitiveId": false,
    "@aws-cdk/aws-cloudfront:defaultSecurityPolicyTLSv1.2_2021": false,
    "@aws-cdk/aws-rds:lowercaseDbIdentifier": false,
    "@aws-cdk/core:stackRelativeExports": false,
    "tooling_region": "eu-west-1",
    "DeploymentEnvironments": [
        {
            "envname": "dev",
            "account": "000000000000",
            "region": "eu-west-1",
            "custom_waf_rules": {
                "allowed_geo_list": [
                  "US",
                  "CN"
                ],
                "allowed_ip_list": [
                  "192.0.2.44/32",
                  "192.0.2.0/24",
                  "192.0.0.0/16"
                ]
              }
        }
    ]
  }
}
```

Use this dataset in an analysis (check the docs customization of analyses) and publish it as a dashboard (docs in publish dashboards)

> Not only RDS
>
> With Quicksight you can go one step further and communicate with other AWS services and data sources. Explore the documentation for cost analyses in AWS with Quicksight or AWS CloudWatch Logs collection and visualization with Quicksight.

5) Bring your dashboard back to data.all Once your dashboard is ready, copy its ID (you can find it in the URL as appears in the below picture)

Back in the data.all Monitoring tab, introduce this dashbaord ID. Now, other tenants can see your dashboard directly from data.all UI!

platform environment. In short, only the role of team A in the data science environment is able to read mydpdata table (in addition to data platform team of course). This is a read-only access, and the data is not moved from the data platform environment to the data science environment.

You can repeat the same thing to check that team B does not have access to the data. Log into data.all with a user in TeamB and select the data science environment. Under the Teams tab, click on the AWS logo to connect to the AWS console assuming the role of TeamB. Go to the Athena Query Editor and you will see that you won't be able to see data shared with team A.

At this stage of the guide, you should better understand how data sharing cross account works. The graph below illustrates where we are in the original scenario.

6. Create a Dataset managed by team A in the data science account

In the previous section of this guide, you went through an example of how you can share data across environment and teams. In this section, we will focus on the creation of datasets in a single account. Team A will create a dataset in the data science environment. We will make sure that other teams invited to the data science environment (t eamB) are not able to access the dataset of team A.

Open data.all with a user in TeamA. In the Contribute section, create a new dataset in the data science environment. Make sure that TeamA owns this dataset.

When the dataset is fully created, upload a csv file from the upload tab of the dataset. Upload this file under a prefix named datateama to create a new Glue table with the same name. After uploading the file, wait a few minutes to let the crawler do its job. In the Tables section, click on Synchronize to display your new table.

Now that the data is uploaded, team A is able to access the data as it is registered as the owner of the dataset. However, team B is not able to read the data even if it has access to the environment. If you log into data.all with the user in team B, you won't be able to see the TeamADataset in the Contribute section. In addition, you will find below two screenshot of the Athena console. In the first screenshot, we assume the role of TeamA in the data science environment (process already explained in the previous section). In the second screenshot, we assume the role of TeamB in the data science environment. When assuming the role of team A, we can see the team A dataset in the database section, and also the datateama table. We can then query the data with Athena. However, when assuming the role of team B in the data science environment, we are not able to see any dataset. This is because in this guide, we have not created or shared any dataset with team B. Team B is thus unable to query the data of team A.

This last section illustrated how you can use teams to manage data access in a single environment. You have reached the end of the guide that illustrated some capabilities that data.all brings. Now that you got the basis, fell free to explore all the other things you can do with your data.

Cleanup

When you are done with this guide, you delete your data.all resources (dataset, environment, organization). This also automatically deletes the Cloudformation stacks created in your AWS accounts.