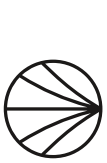
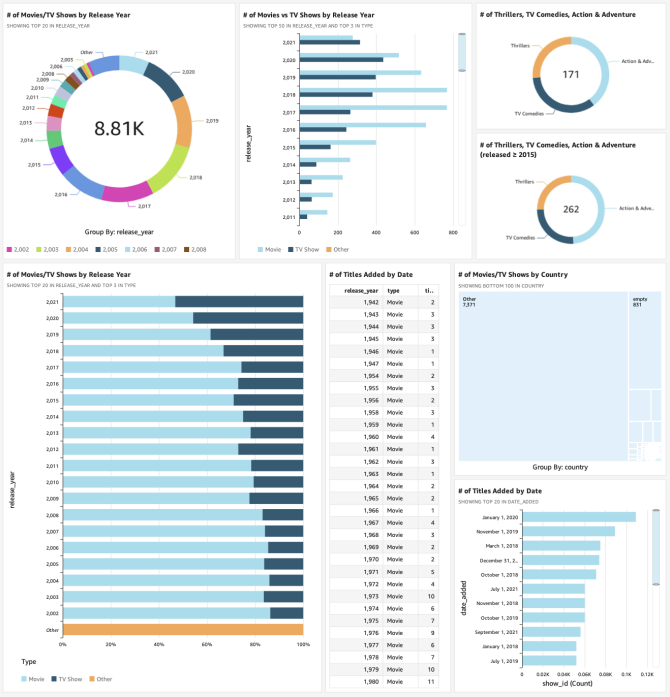


nextwork.org

Visualize data with QuickSight

Axl Cuyugan



Axl Cuyugan
NextWork Student

Introducing Today's Project!

Tools and concepts

Services I used were AWS CDK, S3, and QuickSight. Key concepts I learnt include connecting S3 to QuickSight using the L1 construct, and the importance of setting permissions for the integration to work correctly.

Project reflection

This project took me about 16 hours. The most challenging part was configuring the setup, but it was most rewarding to successfully integrate the dataset from S3 into QuickSight using AWS CDK.

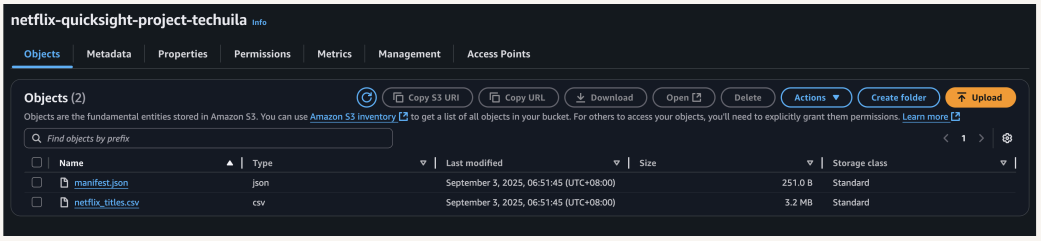
After this project, I plan to focus on cloud security with AWS IAM to deepen my understanding of permissions and access control. I will begin this new project tomorrow.

Axl Cuyugan
NextWork Student

Upload project files into S3

In this project, S3 is used to store two files: manifest.json, which QuickSight uses to define the global settings of the CSV, and netflix_titles.csv, which contains the Netflix dataset.

I edited the manifest.json file to update the S3 bucket location. This step is important because QuickSight relies on this file to know which S3 bucket to use in our AWS account, so we changed the bucket name to ensure it is unique.

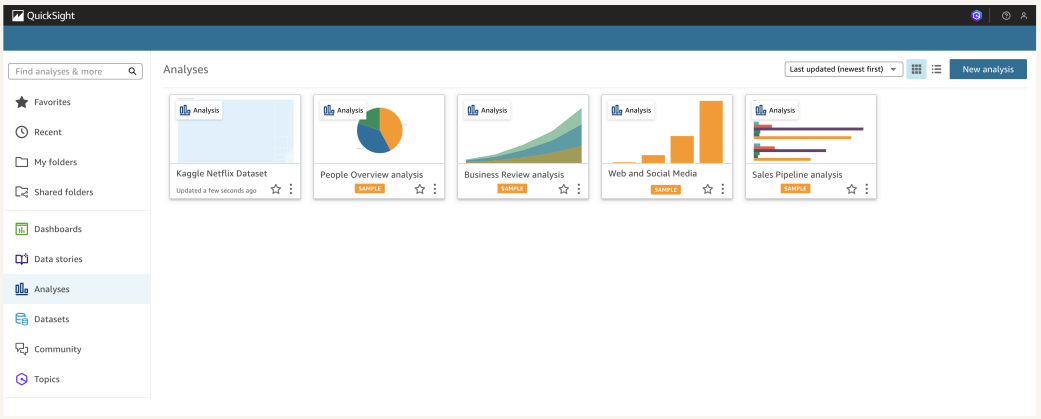


Axl Cuyugan
NextWork Student

Create QuickSight account

Creating a QuickSight account incurs no cost during the free trial period. However, once the trial ends, charges will apply based on the selected pricing plan and usage.

Creating an account took me within a minute.



Axl Cuyugan
NextWork Student

Download the Dataset

I connected the S3 bucket to QuickSight by specifying the bucket name and the manifest file key from the S3 parameters.

The manifest.json guides QuickSight to the dataset in S3. We also set uploadSettings in CfnDataSet to define file format and structure, ensuring QuickSight interprets the data correctly.

```
  public static QUICKSIGHT_DATASET_NAME = 'Kaggle Netflix Dataset'

  constructor(scope: Construct, id: string, props: StackProps & { username: string }) {
    super(scope, id, props)

    const { bucket, deploy } = new S3BucketAndDeploy(this, 'S3BucketAndDeploy', {
      name: NetflixQuicksightStack.BUCKET_NAME,
      deploySources: [Source.asset(path.join(__dirname, '..', 'data/netflix-dataset'))]
    })

    const quicksight = new Quicksight(this, 'NetflixQuicksight', {
      prefix: 'netflix',
      datasourceProps: {
        name: 'Netflix S3 Data Source',
        s3Parameters: {
          manifestFileLocation: {
            bucket: bucket.bucketName,
            key: NetflixQuicksightStack.MANIFEST_KEY
          }
        }
      },
      datasetProps: {
        name: NetflixQuicksightStack.QUICKSIGHT_DATASET_NAME,
        inputColumns: [
          { name: 'show_id', type: 'STRING' },
          { name: 'type', type: 'STRING' },
          { name: 'title', type: 'STRING' },
          { name: 'director', type: 'STRING' },
          { name: 'cast', type: 'STRING' },
          { name: 'country', type: 'STRING' },
          { name: 'date_added', type: 'STRING' },
          { name: 'release_year', type: 'STRING' },
          { name: 'rating', type: 'STRING' },
          { name: 'duration', type: 'STRING' },
          { name: 'listed_in', type: 'STRING' },
          { name: 'description', type: 'STRING' }
        ],
        dataTransforms: [
          {
            castColumnTypeOperation: {
              columnName: 'release_year',
              newColumnType: 'INTEGER'
            }
          }
        ]
      },
      managedPolicyProps: {
        bucket,
        deployment: deploy,
        quicksightAccountArn: `arn:aws:quicksight:${this.region}:${this.account}:user/default/${props.use
      }
    })

    quicksight.datasource.node.addDependency(quicksight.managedPolicy)
    quicksight.datasource.node.addDependency(deploy)
  }
```

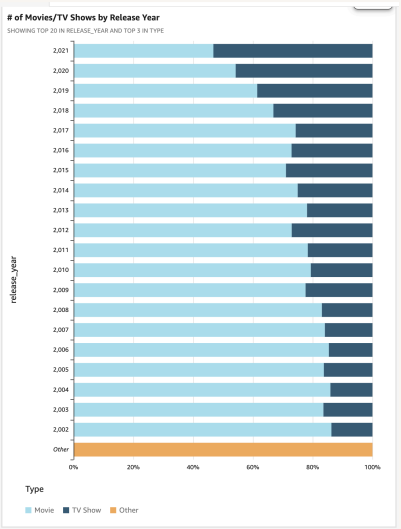
Axl Cuyugan
NextWork Student

My first visualization

To create visualizations on QuickSight, I select a dataset, open a new analysis, and drag fields into the visualization pane. Then, I choose a chart type, customize its properties, and optionally combine multiple visuals into a dashboard.

The chart shown here is a breakdown of the number of Movies, TV Shows, and Others by release year. This allows us to compare how the distribution of titles changes over time.

I created this graph by dragging release_year to the Y-axis and type to the color grouping, using the count of titles as the metric to show Movies, TV Shows, and Others by release year.

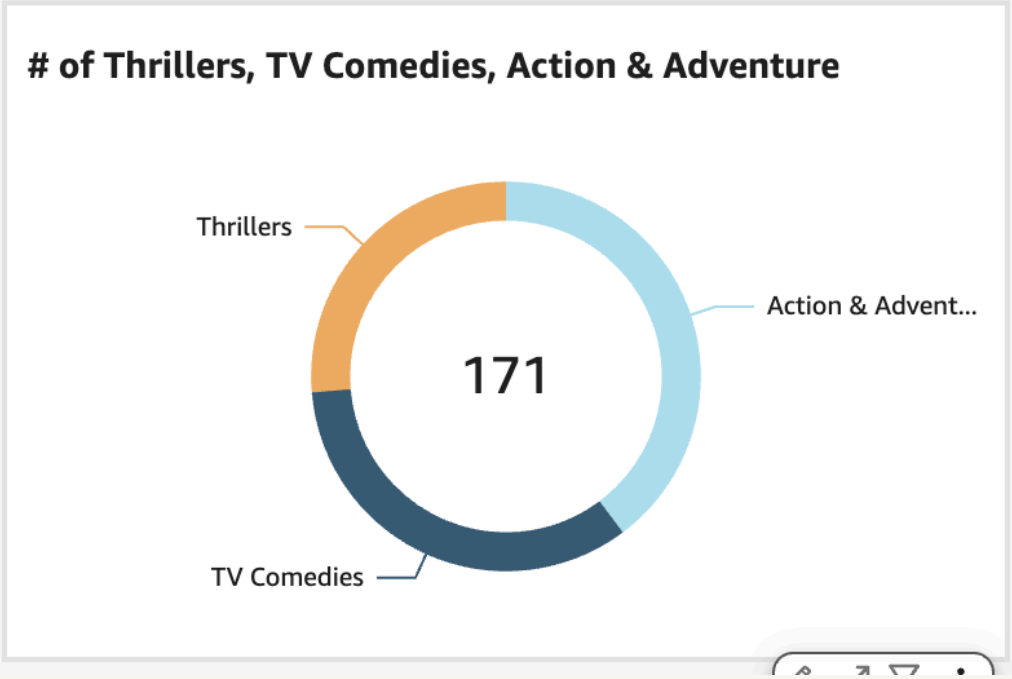


Axl Cuyugan
NextWork Student

Using filters

Filters are useful for narrowing down data to specific conditions, making it easier to focus on relevant insights, compare subsets, and answer targeted questions within a visualization or dashboard.

This visualization is a breakdown of the number of titles by genre. I applied a filter to show only Thrillers, Action & Adventure, and TV Comedies, displayed in a donut chart for clear comparison.

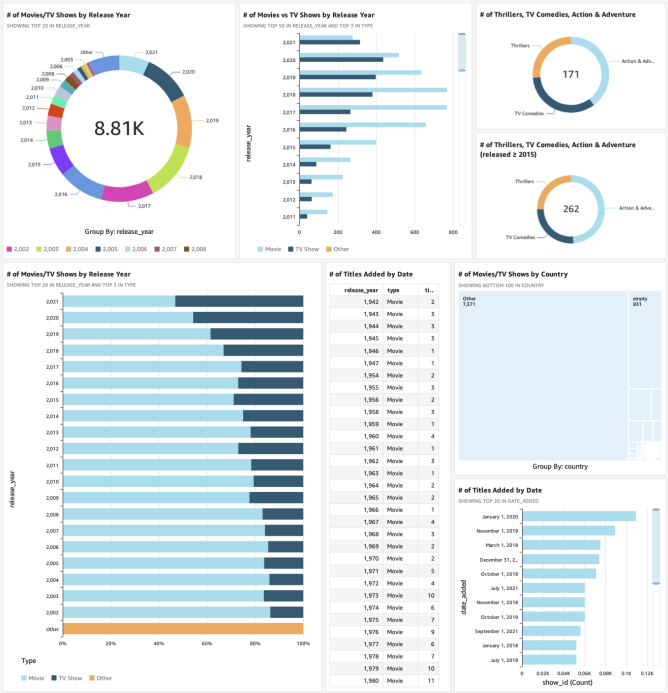


Axl Cuyugan
NextWork Student

Setting up a dashboard

As a finishing touch, I arranged my panels to match the example layout, ensuring a clean, organized dashboard that makes the visualizations easier to read and interpret.

Did you know you could export your dashboard as PDFs too? I did this by opening the dashboard, clicking Share → Export to PDF, and QuickSight generated a downloadable file with all my visualizations.



The place to learn & showcase your skills

Check out nextwork.org for more projects