

Introduction to Cube Cloud



Community Code of Conduct

- We want to foster an open and welcoming environment where everyone feels they belong in the Cube.js community.
- The full text of our Code of Conduct is available at https://github.com/cube-js/cube.js/blob/master/CODE_OF_CONDUCT.md
- Any instances of inappropriate/unacceptable behavior can be reported to conduct@cube.dev.

Some quick notes

- If you have any questions during the workshop, please feel free to type them in the “Q&A” (on Zoom & YouTube LiveStream).
- First Cube Cloud-focused workshop!
- Recording of today’s workshop will be posted on the Cube Dev YouTube channel.
- All attendees will receive a post-event survey and we’d appreciate your feedback to help us with future events.
 - You will have a new pair of Cube socks when you submit your survey! 

Upcoming workshops

- Next Cube Community Call in September 2022
 - Date: September 14th
 - Topic: Community & Product updates
 - Details at <https://cube.dev/events/cube-community-call-in-september-2022>
- Next workshop with our friends from Firebolt
 - Date: September 15th
 - Topic: Building blazing-fast data apps with Cube and Firebolt
 - You can register at <https://cube.dev/events/firebolt-webinar-data-apps>

Previous workshops

- Building Data Apps
 - <https://cube.dev/events/building-data-app>
- Cube SQL API and BI tool support
 - <https://cube.dev/events/sql-api>
- Connecting BI tools to Cube
 - <https://cube.dev/events/adv-sql-api>

What we will discuss today

- The goal of this workshop
- Introduction to Cube Cloud
- Hands-on lab with Cube Cloud
 - Dev tooling, Observability, Alerting, Performance
- Q & A

Overview of prerequisites

Prep Guide and Code

- Prep Guide

<https://descriptive-reply-0b7.notion.site/Workshop-Introduction-to-Cube-Cloud-Prep-Guide-cf2a3c6d332c41ba98afdb8e3911ff17>

- Code

<https://descriptive-reply-0b7.notion.site/Workshop-Introduction-to-Cube-Cloud-Code-88c77805564d460e80eca4bfc41ddca0>

The goal of this workshop

Learn how to

- Build, deploy, and scale Cube apps faster
- Run Cube apps in production
- Keep developer productivity high
- Observability
 - Alerts when issues happen
 - Metrics to see why issues happen
- Improve performance

Introduction to Cube Cloud

Introduction to Cube Cloud

- Cube Cloud & open-source Cube – Same features
- Fastest way to develop Cube apps
- Deploy, scale, optimize, and run Cube apps in production

Introduction to Cube Cloud

- Managed infrastructure
 - scales instances, memory, high-availability, pre-aggregations management, caching, and scalability, etc.
- Just like Serverless or managed K8s 🤔

Overview of the **TPC-H** dataset

Sample e-commerce data

- Relatable
- Applicable for various use-cases
- Used for benchmarks
 - <https://www.tpc.org/tpch/default5.asp>
- Sample dataset available in Snowflake
 - <https://docs.snowflake.com/en/user-guide/sample-data-tpch.html>

Demo 0

Introduction to Cube Cloud

Hands-on lab with Cube Cloud

Part 1

Dev tooling

Dev tooling

- Developer experience in Cube Cloud
- Developer productivity 

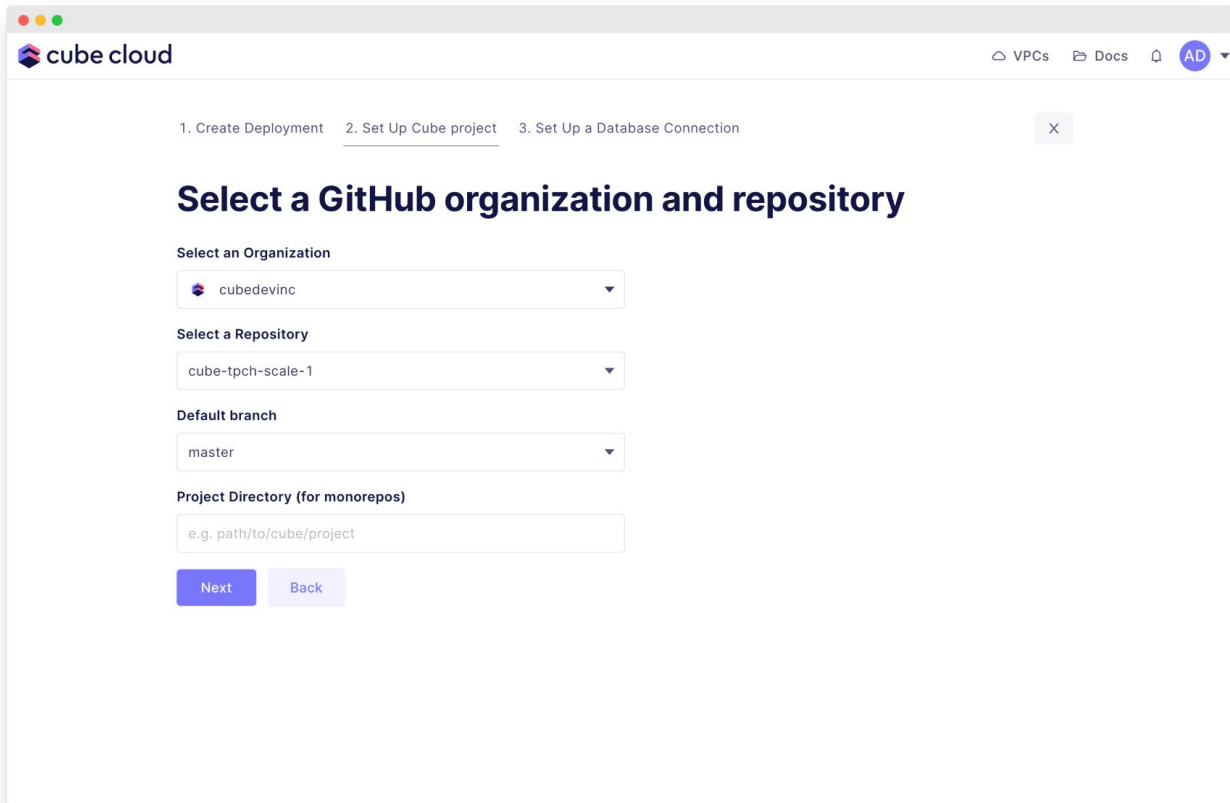
Dev tooling

- Migrate a Cube app to Cube Cloud
- Automating deployments
- Version control
- Dev mode

Part 1.1

Migrate a Cube app to Cube Cloud

Set up a Cube app with GitHub



The screenshot shows the 'cube cloud' web interface. At the top, there's a navigation bar with 'VPCs', 'Docs', and a user profile icon labeled 'AD'. Below this, a progress bar indicates three steps: '1. Create Deployment', '2. Set Up Cube project' (which is the current step), and '3. Set Up a Database Connection'. The main heading is 'Select a GitHub organization and repository'. There are four dropdown menus: 'Select an Organization' with 'cubedevinc' selected, 'Select a Repository' with 'cube-tpch-scale-1' selected, 'Default branch' with 'master' selected, and 'Project Directory (for monorepos)' with the placeholder text 'e.g. path/to/cube/project'. At the bottom, there are 'Next' and 'Back' buttons.

cube cloud

VPCs Docs AD

1. Create Deployment 2. Set Up Cube project 3. Set Up a Database Connection X

Select a GitHub organization and repository

Select an Organization

cubedevinc

Select a Repository

cube-tpch-scale-1

Default branch

master

Project Directory (for monorepos)

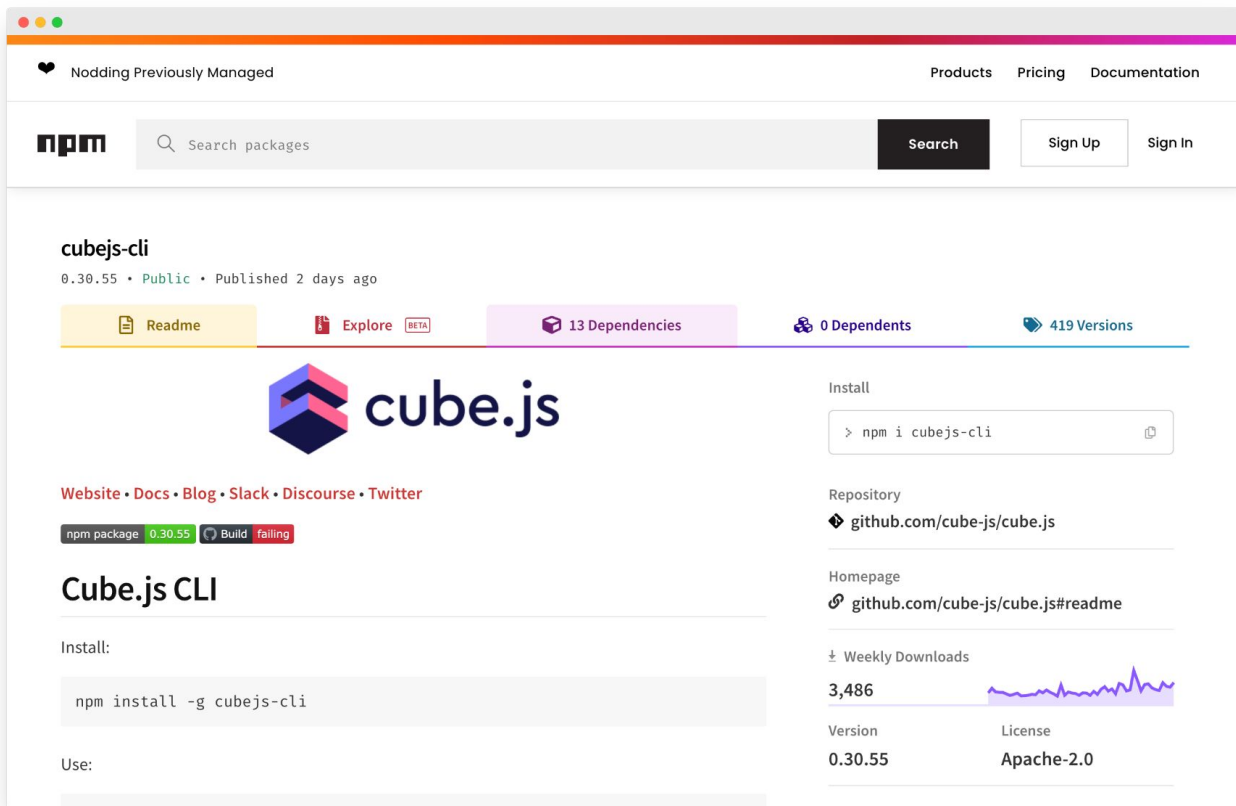
e.g. path/to/cube/project

Next Back

Set up a Cube app with GitHub

- Grabs `cube.js` file and schema files

Set up a Cube app with Cube's CLI



The screenshot shows the npm package page for **cubejs-cli**. The page header includes the npm logo, a search bar, and links for Products, Pricing, and Documentation. The package name **cubejs-cli** is displayed with its version **0.30.55**, status **Public**, and publication time **2 days ago**. Below this, there are links for **Readme**, **Explore** (marked BETA), **13 Dependencies**, **0 Dependents**, and **419 Versions**. The main section features the **cube.js** logo and a list of links: **Website**, **Docs**, **Blog**, **Slack**, **Discourse**, and **Twitter**. A status bar indicates the package is **npm package 0.30.55** with a **Build failing** warning. The **Cube.js CLI** section provides an **Install:** command: `npm install -g cubejs-cli`. The **Repository** section points to **github.com/cube-js/cube.js**, and the **Homepage** section points to **github.com/cube-js/cube.js#readme**. The **Weekly Downloads** section shows a graph and the number **3,486**. The **Version** and **License** section shows **0.30.55** and **Apache-2.0** respectively.

Nodding Previously Managed

Products Pricing Documentation

npm Search packages Search Sign Up Sign In

cubejs-cli
0.30.55 • Public • Published 2 days ago

Readme Explore BETA 13 Dependencies 0 Dependents 419 Versions

 **cube.js**

Website • Docs • Blog • Slack • Discourse • Twitter

npm package 0.30.55 Build failing

Cube.js CLI

Install:

```
npm install -g cubejs-cli
```

Use:

Install

```
> npm i cubejs-cli
```

Repository

github.com/cube-js/cube.js

Homepage

github.com/cube-js/cube.js#readme

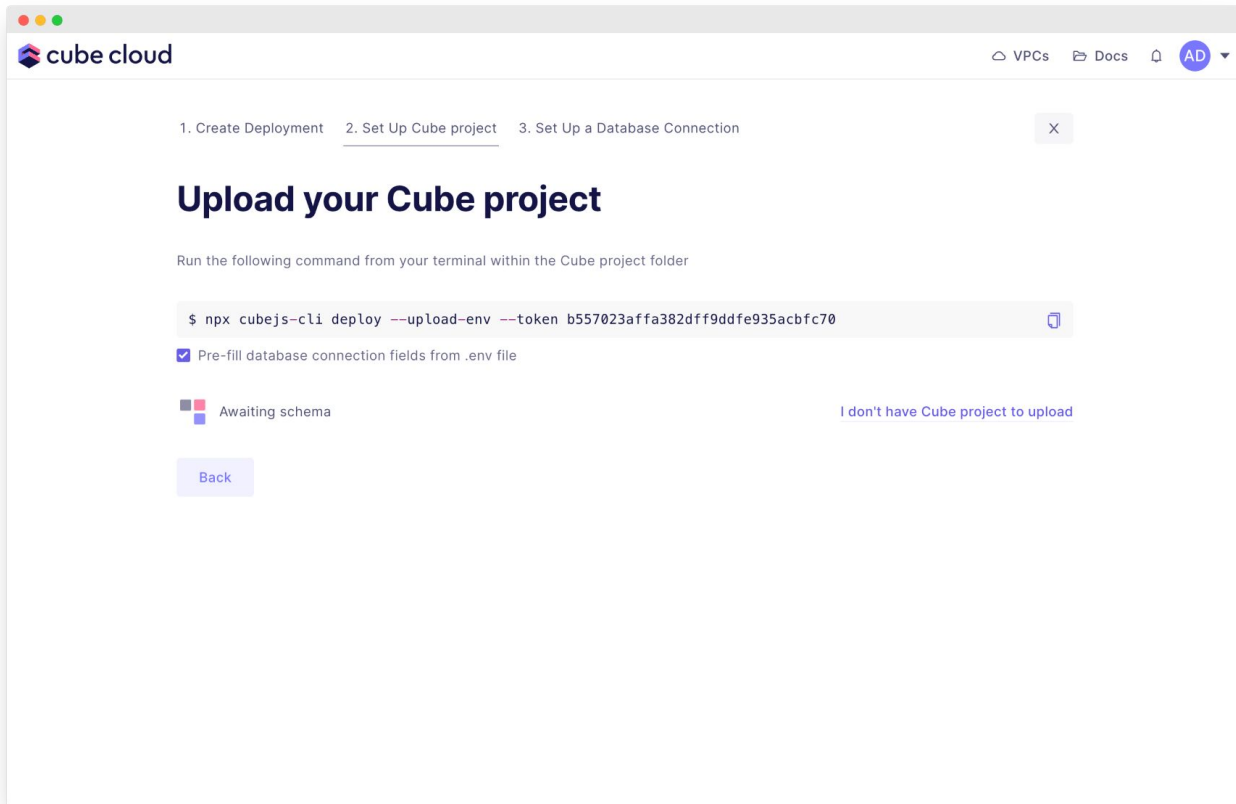
Weekly Downloads

3,486

Version License

0.30.55 Apache-2.0

Set up a Cube app with Cube's CLI



The screenshot shows the 'cube cloud' web interface. At the top, there's a navigation bar with 'VPCs', 'Docs', and a user profile icon labeled 'AD'. Below this is a progress bar with three steps: '1. Create Deployment', '2. Set Up Cube project' (which is underlined), and '3. Set Up a Database Connection'. A close button 'X' is on the right of the progress bar.

Upload your Cube project

Run the following command from your terminal within the Cube project folder

```
$ npx cubejs-cli deploy --upload-env --token b557023affa382dff9ddfe935acbfc70
```

☒ Pre-fill database connection fields from .env file

☐ Awaiting schema [I don't have Cube project to upload](#)

[Back](#)

Set up a Cube app with Cube's CLI

```
npm i --save-dev cubejs-cli  
npm i --save-dev @cubejs-backend/server  
npx cubejs-cli deploy --upload-env --token <YOUR_TOKEN>
```

Set up a Cube app with Cube's CLI

- Creates a `.cubecloud` file
- Grabs `cube.js` file and schema files
- Loads environment variables from the `.env` file

Part 1.2

Automating deployments

Cube's CLI deployment

The screenshot shows the Cube Cloud web interface. The top navigation bar includes the 'cube cloud' logo, the path 'Deployments / adnan-test-3', and links for 'VPCs', 'Docs', and a user profile 'AD'. A purple button labeled '</> Enter Development Mode' is on the left. A status bar indicates 'API is ready (Build #1974)' with a 'Copy API URL' link. The left sidebar contains a menu with 'Overview', 'Playground', 'Schema', 'Queries', 'Pre-Aggregations', 'Metrics', and 'Settings' (highlighted). The main content area is titled 'Settings' and has tabs for 'General', 'Configuration', and 'Build & Deploy' (selected). Under 'Build & Deploy', there are two sections: 'Deploy manually with Git' and 'Deploy with'. The 'Deploy manually with Git' section shows a 'Deploy Token' field with the value 'fb153ad68fe20100280c405cd76a5c7c'. Below it, text explains how to upload the Cube config and data schema using the CLI. A code block shows the command: `$ npx cubejs-cli deploy --token fb153ad68fe20100280c405cd76a5c7c`. Further down, text mentions using the CLI for continuous deployment with a link to 'Learn more about continuous deployment in docs'. A YAML configuration for GitHub Actions is shown in a code block:

```
name: My Cube App
on:
  push:
    paths:
      - '**'
    branches:
      - 'master'
jobs:
  deploy:
    name: Deploy My Cube App
    runs-on: ubuntu-latest
    timeout-minutes: 30
    steps:
      - name: Checkout
        uses: actions/checkout@v2
      - name: Use Node.js 14.x
        uses: actions/setup-node@v1
```

 The 'Deploy with' section has radio buttons for 'Deploy with CLI' (selected) and 'Deploy with GIT', and an 'Edit' button.

cube cloud Deployments / adnan-test-3 VPCs Docs AD

</> Enter Development Mode API is ready (Build #1974) Copy API URL

Overview Playground Schema Queries Pre-Aggregations Metrics Settings

Settings

General Configuration **Build & Deploy**

Deploy manually with Git

Deploy Token

fb153ad68fe20100280c405cd76a5c7c

Deploy manually with CLI

To upload your Cube config and data schema to the Cube Cloud run the following command in the directory with schema folder and your Cube config:

```
$ npx cubejs-cli deploy --token fb153ad68fe20100280c405cd76a5c7c
```

You can use the CLI to set up continuous deployment for a Git repository. The example below shows the example configuration for [GitHub Actions](#). [Learn more about continuous deployment in docs](#)

```
name: My Cube App
on:
  push:
    paths:
      - '**'
    branches:
      - 'master'
jobs:
  deploy:
    name: Deploy My Cube App
    runs-on: ubuntu-latest
    timeout-minutes: 30
    steps:
      - name: Checkout
        uses: actions/checkout@v2
      - name: Use Node.js 14.x
        uses: actions/setup-node@v1
```

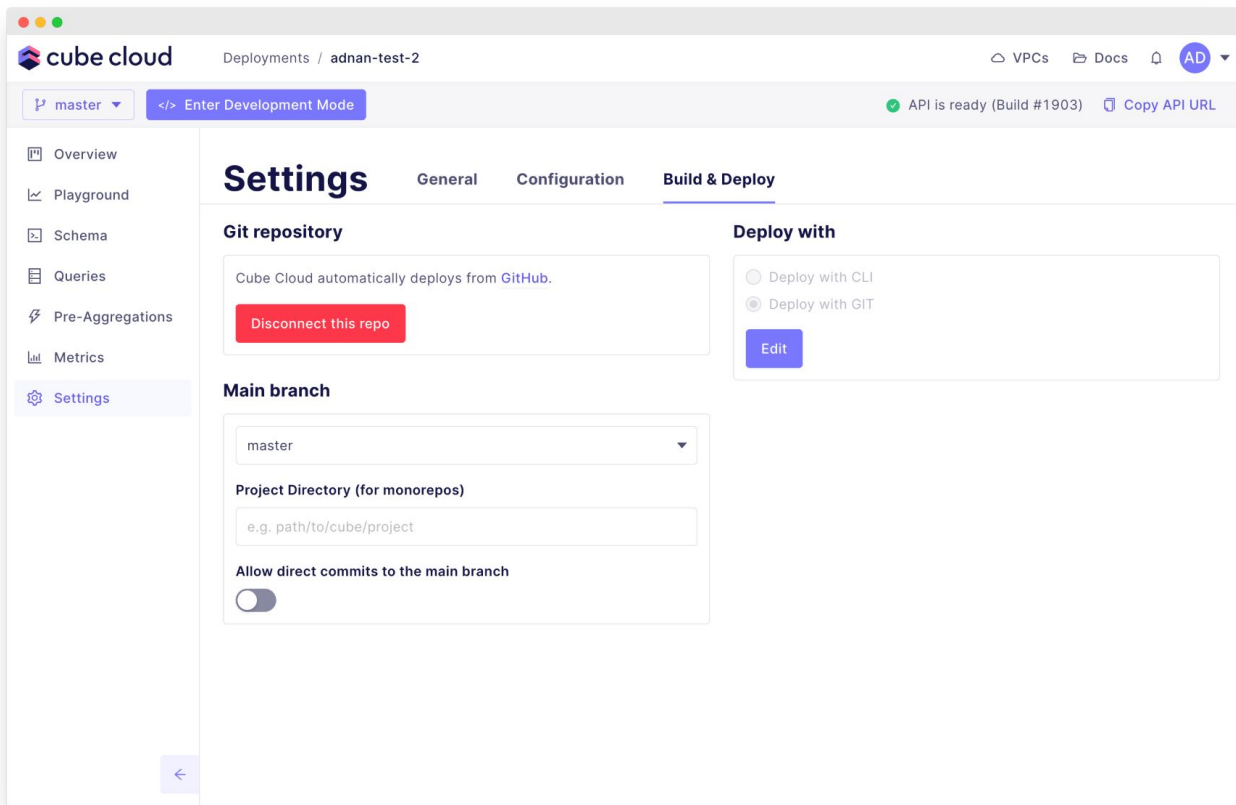
Deploy with

☒ Deploy with CLI

☐ Deploy with GIT

Edit

GitHub deployment



Git deployment

The screenshot shows the Cube Cloud interface for a deployment named 'intro-to-cube-cloud-git'. The 'Build & Deploy' tab is active under the 'Settings' section. The interface includes a sidebar with navigation options like Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The main content area is divided into sections for 'Git repository', 'Main branch', and 'Deploy manually with Git'. The 'Git repository' section has a 'Connect to GitHub' button. The 'Main branch' section shows 'master' as the selected branch and a toggle for 'Allow direct commits to the main branch'. The 'Deploy manually with Git' section provides terminal commands to add the Cube Cloud remote and push to the master branch, along with fields for 'Git User' (negaxhigtekj) and 'Git Password' (znhgewdfjleq).

cube cloud Deployments / intro-to-cube-cloud-git VPCs Docs AD

master </> Enter Development Mode API is ready (Build #48) Copy API URL

Settings

General Configuration **Build & Deploy**

Git repository

You can connect your Git repository to Cube Cloud to configure continuous deployment.

Connect to GitHub

Deploy with

☐ Deploy with CLI
☒ Deploy with GIT

Edit

Main branch

master

Allow direct commits to the main branch

Deploy manually with Git

Add Cube Cloud remote to your git repository and push it:

```
$ git config credential.helper store  
$ git remote add cubecloud https://workshops.cubecloud.com  
$ git push cubecloud master
```

Use your generated git credentials to authenticate git operations.

Git User
negaxhigtekj

Git Password
znhgewdfjleq

Automating deployments

- Cube's CLI
 - Automated deployment with GitHub actions
- GitHub
 - Auto-deploy
- Git
 - Manual deployment

Demo 1

Migration and Deployment Automation

Part 1.3

Version control

Version control in Cube Cloud

- We want you to keep using your favorite tools
- **How?** You get...
 - Development environment you're used to
 - Branching like you're used to with Git

Version control in Cube Cloud

The screenshot displays the Cube Cloud web interface for a deployment named 'adnan-test-4'. The interface is divided into several sections:

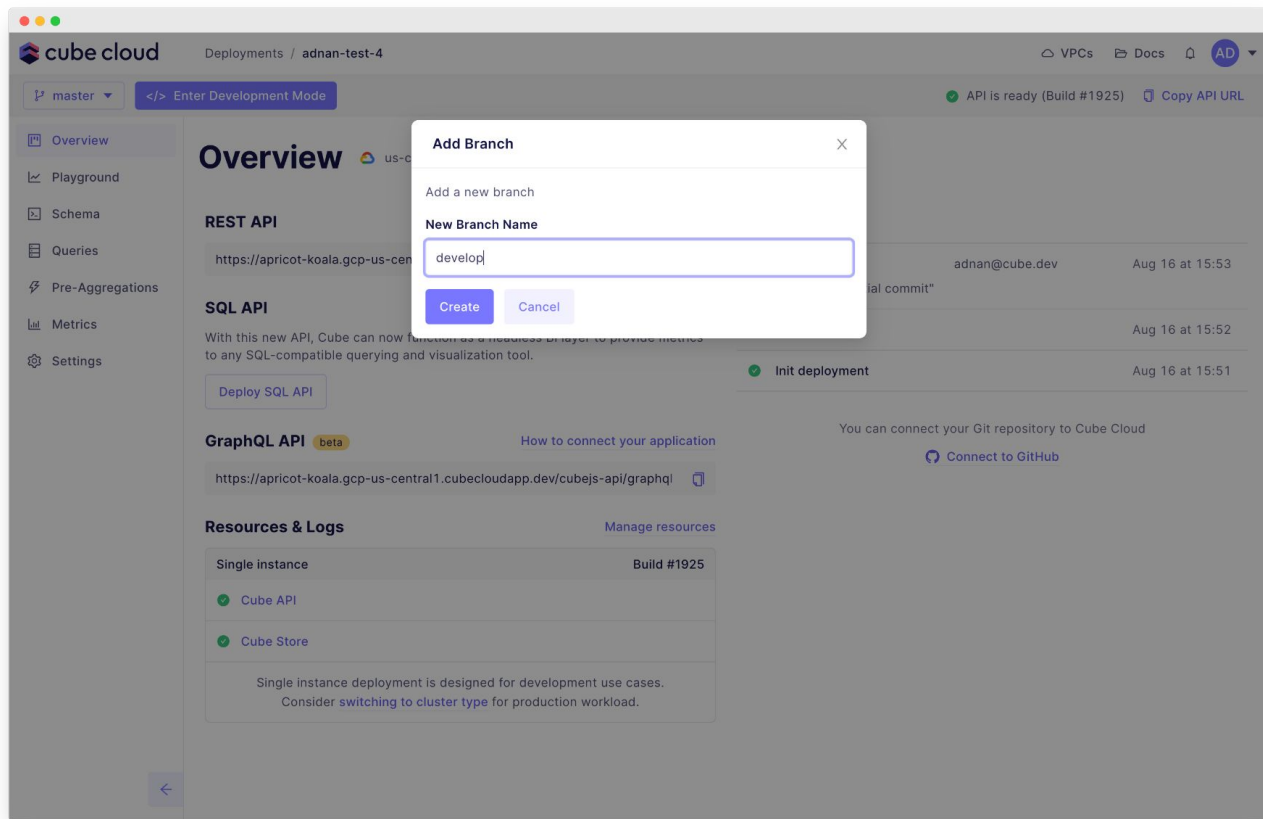
- Header:** Includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'.
- Toolbar:** Features a 'master' branch selector, a button to 'Enter Development Mode', and a status indicator 'API is ready (Build #1925)' with a 'Copy API URL' link.
- Left Sidebar:** Contains navigation links for 'Playground', 'Schema', 'Queries', 'Pre-Aggregations', 'Metrics', and 'Settings'.
- Overview Section:** Shows the deployment location 'us-central1 (Iowa)' and an 'Add Branch' button.
- REST API:** Provides the API endpoint 'https://apricot-koala.gcp-us-central1.cubecloudapp.dev/cubejs-api/v1' and a 'Deploy SQL API' button.
- SQL API:** Describes the SQL API as a headless BI layer and includes a 'Deploy SQL API' button.
- GraphQL API:** Marked as 'beta', it provides the endpoint 'https://apricot-koala.gcp-us-central1.cubecloudapp.dev/cubejs-api/graphql' and a 'Deploy GraphQL API' button.
- Activity Log:** A table listing recent events:

Event	User	Time
Build #1925	adnan@cube.dev	Aug 16 at 15:53
8fb46eb0: "Initial commit"		
Env var update		Aug 16 at 15:52
Init deployment		Aug 16 at 15:51
- Resources & Logs:** A table showing the deployment configuration:

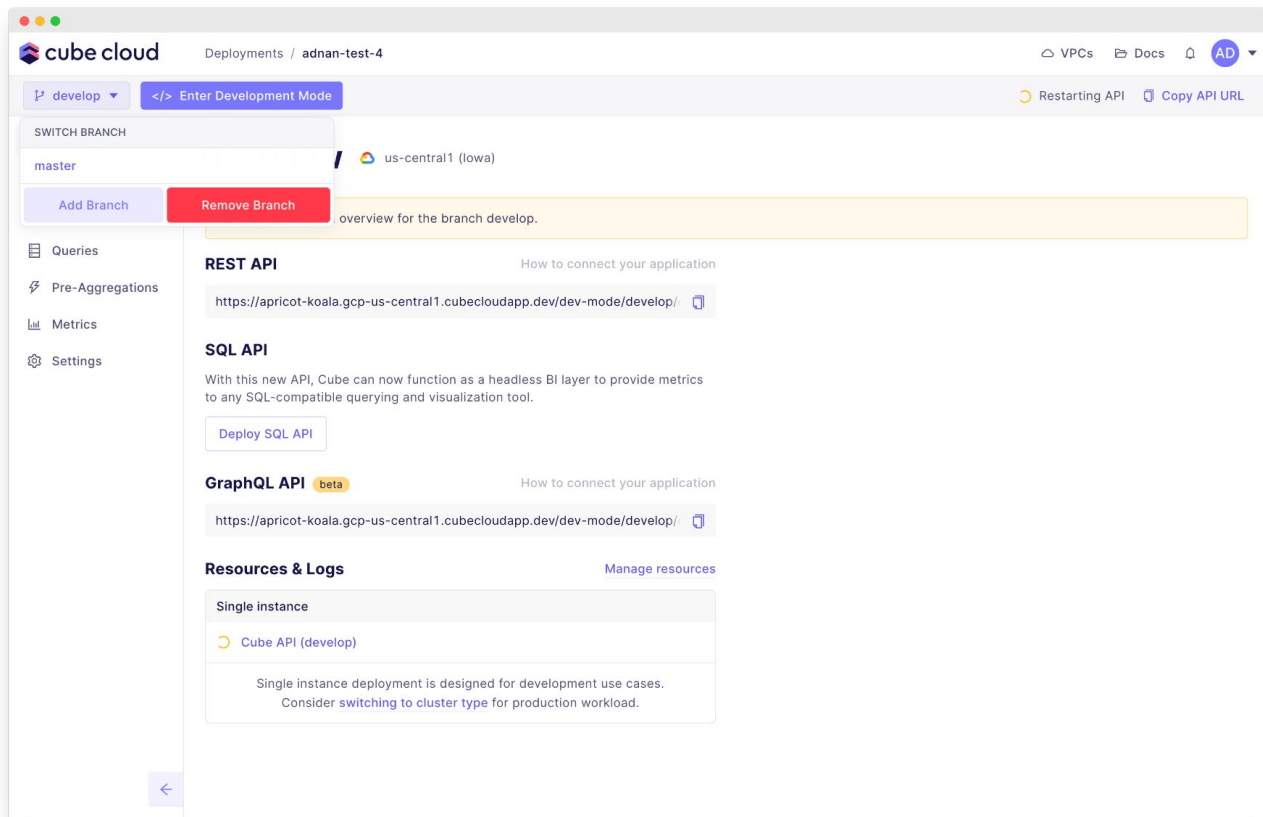
Resource	Build
Single instance	Build #1925

Below the table, it lists 'Cube API' and 'Cube Store' as resources. A note states: 'Single instance deployment is designed for development use cases. Consider switching to cluster type for production workload.'
- Footer:** A message 'You can connect your Git repository to Cube Cloud' with a 'Connect to GitHub' button.

Version control in Cube Cloud



Version control in Cube Cloud



Version control in Cube Cloud

- Select which branch is the main branch
- Allow direct commits to the main branch
 - Disabled by default

Version control in Cube Cloud

The screenshot shows the Cube Cloud web interface. At the top, the header includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'. Below the header, a dark blue bar contains 'Development Mode', 'Exit Dev Mode', a branch selector set to 'develop', and status indicators 'API is ready' and 'Copy API URL'.

The left sidebar lists navigation options: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings (which is highlighted).

The main content area is titled 'Settings' and has three tabs: 'General', 'Configuration', and 'Build & Deploy' (which is active). The 'Build & Deploy' tab contains three sections:

- Git repository:** A text box stating 'You can connect your Git repository to Cube Cloud to configure continuous deployment.' with a 'Connect to GitHub' button.
- Deploy with:** Two radio buttons, 'Deploy with CLI' and 'Deploy with GIT' (which is selected), with an 'Edit' button.
- Main branch:** A dropdown menu showing 'master' and a toggle switch for 'Allow direct commits to the main branch' which is currently turned off.

Below these sections is the 'Deploy manually with Git' section, which includes instructions to add the Cube Cloud remote and push. It provides a code block with the following commands:

```
$ git config credential.helper store
$ git remote add cubecloud https://examples.cubecloud.com
$ git push cubecloud master
```

At the bottom of this section is a 'Generate Git credentials' button. A back arrow is visible in the bottom left corner of the settings area.

Part 1.4

Dev Mode

Dev Mode

- Separate environment for testing changes
- Does not interfere with the production API

Dev Mode

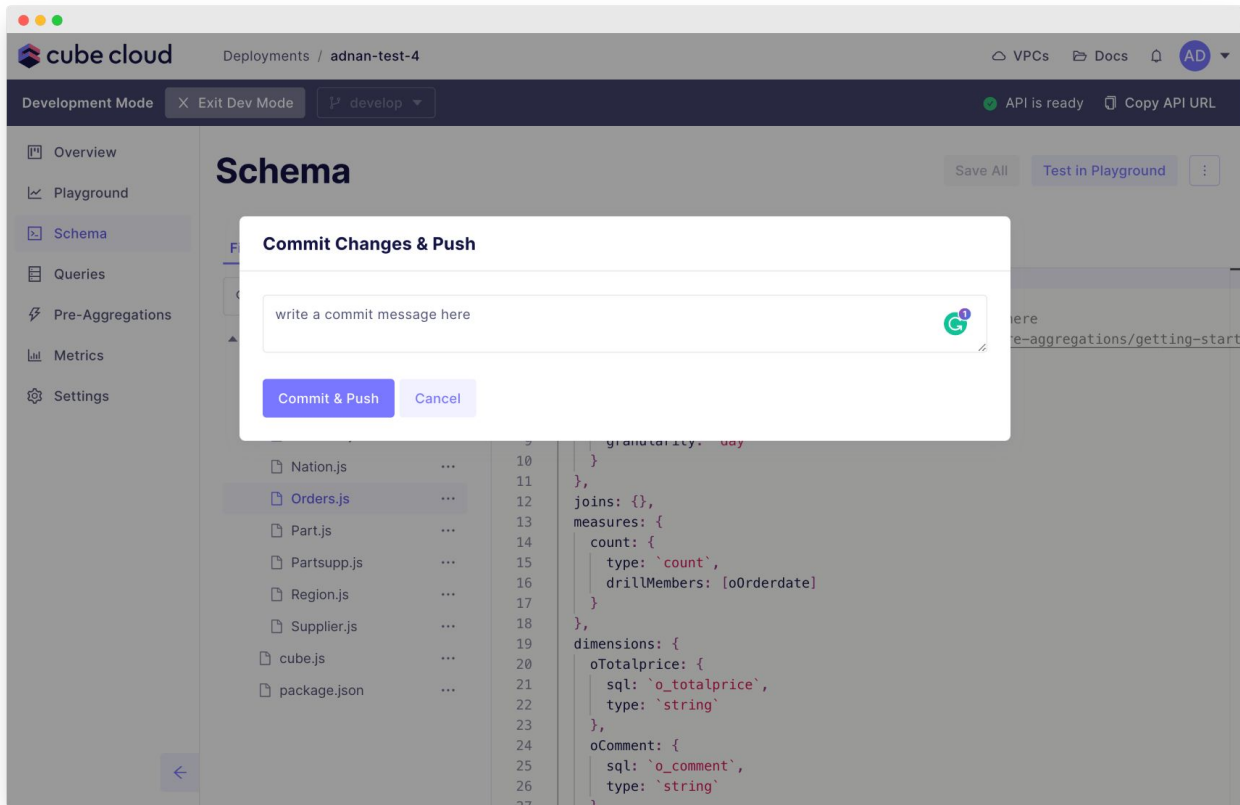
The screenshot displays the Cube Cloud web interface in 'Development Mode'. The top navigation bar includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile. Below this, a dark blue bar shows 'Development Mode' with an 'Exit Dev Mode' button and a 'develop' branch selector. On the right of this bar, it indicates 'API is ready' and provides a 'Copy API URL' button.

The left sidebar contains a menu with 'Overview', 'Playground', 'Schema' (selected), 'Queries', 'Pre-Aggregations', 'Metrics', and 'Settings'. The 'Schema' section is active, showing a file explorer with a search bar and a list of files: 'Customer.js', 'Lineitem.js', 'Nation.js', 'Orders.js' (selected), 'Part.js', 'Partsupp.js', 'Region.js', 'Supplier.js', 'cube.js', and 'package.json'.

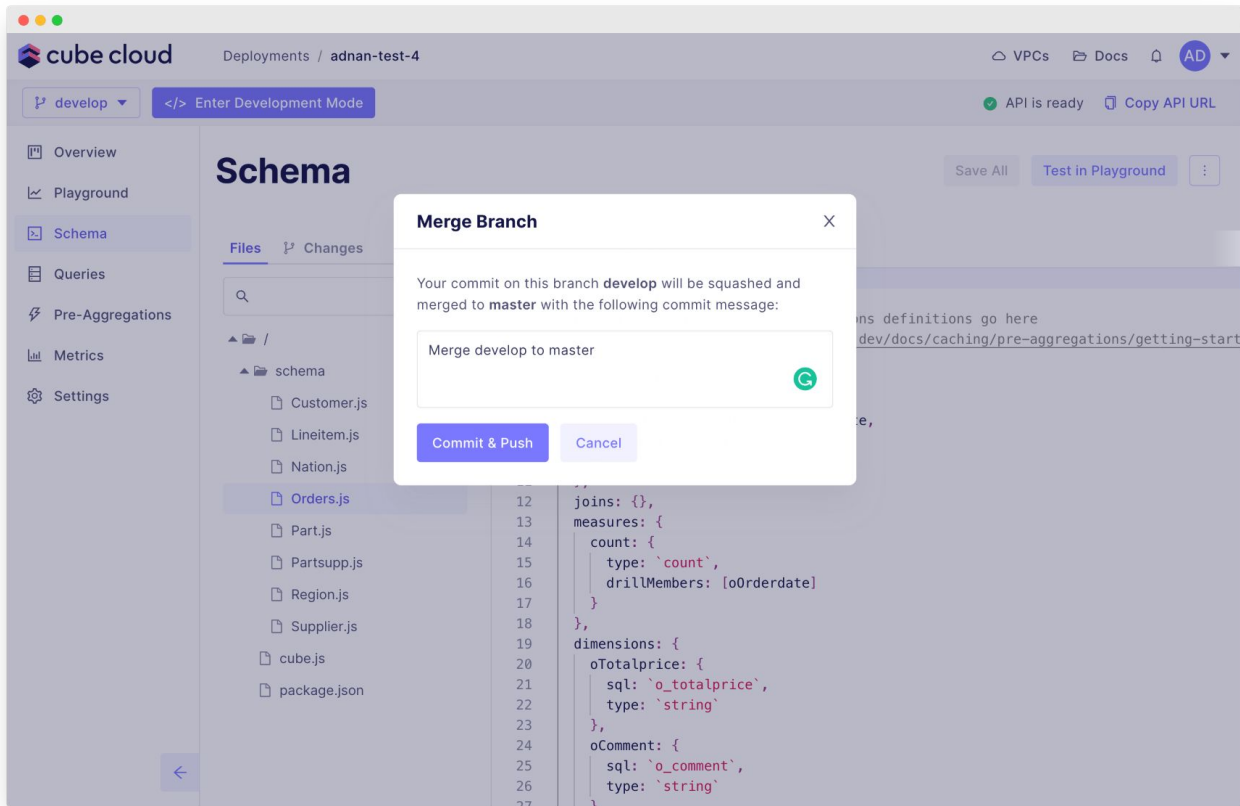
The main area is titled 'Schema' and shows the content of 'Orders.js'. The code defines a cube for 'Orders' with a SQL query, pre-aggregations, and measures. A right-hand menu is open, showing 'GIT ACTIONS' (Commit and push, Revert to production, Rebase onto develop, Rebase onto master) and 'SCHEMA ACTIONS' (Generate schema, Add rollup).

```
1 cube('Orders', {
2   sql: 'SELECT * FROM public.orders',
3   preAggregations: { // Pre-Aggregations definitions go here
4     // Learn more here: https://cube.dev/docs/caching/pre-
5   },
6   ordersByDay: {
7     measures: [Orders.count],
8     timeDimension: Orders.oOrderdate,
9     granularity: 'day'
10  },
11 },
12 joins: {},
13 measures: {
14   count: {
15     type: 'count',
16     drillMembers: [oOrderdate]
17   },
18 },
19 dimensions: {
20   oTotalprice: {
21     sql: 'o_totalprice',
22     type: 'string'
23   },
24   oComment: {
25     sql: 'o_comment',
26     type: 'string'
27   }
28 }
```

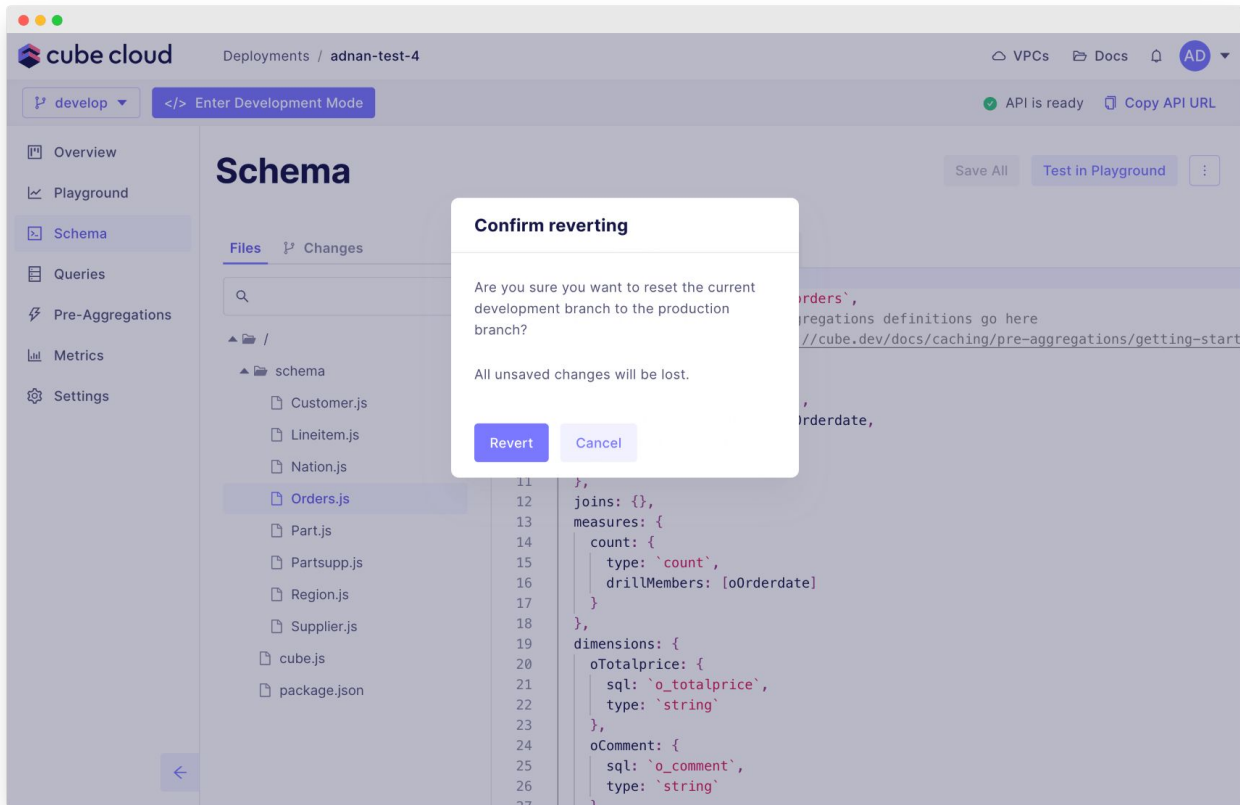
Dev Mode



Dev Mode



Dev Mode



Demo 2

Version control and Dev mode

Part 2

Observability

Observability

- Debugging you're familiar with
- Get insight into metrics to see why issues happen
- Get alerted when issues happen

Observability

- Debugging
- Query tracing and drill-downs
- Monitoring metrics
- Alerting

Part 2.1

Debugging

Debugging config and infra issues

- Activity log
- Resource logs

Activity log

- Version control changes
- Build jobs
- Config changes

Activity log

The screenshot shows the 'cube cloud' web interface. The top navigation bar includes the 'cube cloud' logo, the path 'Deployments / adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below the navigation bar, there's a dropdown menu set to 'master' and a button labeled '</> Enter Development Mode'. On the right side of this bar, a status message says 'API is ready (Build #1933)' with a green checkmark, and a 'Copy API URL' button.

The left sidebar contains a list of navigation items: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The main content area is titled 'Activity Log' with a back arrow. It displays a list of activities, each with a green checkmark icon, a build number, a description, the user 'adnan@cube.dev', and a timestamp.

Build	Description	User	Timestamp
Build #1933	a4555386: Merge develop to master	adnan@cube.dev	Aug 18 at 12:17
Build #1926	fd07cd96: Merge develop to master	adnan@cube.dev	Aug 17 at 12:55
Build #1925	8fb46eb0: "Initial commit"	adnan@cube.dev	Aug 16 at 15:53
Env var update			Aug 16 at 15:52
Init deployment			Aug 16 at 15:51

Activity log

The screenshot displays the 'cube cloud' web interface. At the top, the navigation bar includes the 'cube cloud' logo, the current deployment 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below the navigation bar, there's a section with a 'master' dropdown and a '</> Enter Development Mode' button. On the right of this section, a green status indicator shows 'API is ready (Build #1933)' and a 'Copy API URL' button.

The main content area is titled 'Build Job #1926' with a 'Rerun' button. It features two summary boxes: 'Status' showing 'success' and 'Scheduled' showing 'Aug 17 at 12:55'. Below these is a 'Logs' section with a list of build steps:

```
Cloning into 'app'...
Branch: master
Docker template detected.
yarn install v1.22.15
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
Done in 0.21s.
Preparing archive...
Done. Uploading...
```

A sidebar on the left contains navigation links: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. A back arrow button is located at the bottom left of the main content area.

Resource logs for your infra

- Cube API logs
- Cube refresh worker logs
- Cube Store logs

Cube API logs when deploying new builds

cube cloud

Deployments / adnan-test-4

VPCs Docs AD

master

</> Enter Development Mode

Deploying Build #1935 API is ready (Build #1934) Copy API URL

Overview

Playground

Schema

Queries

Pre-Aggregations

Metrics

Settings

Resources

Cube API

Status 577dd46cb8-6sbgm
running

Status 5877cd774b-wcjmr
running

Status 8476cbc9d5-9t558
waiting: PodInitializing

Main Logs

5877cd774b-wcjmr

2022-08-18T19:37:22.142Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:37:51.821Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:38:22.033Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:38:51.795Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:39:22.014Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:39:51.769Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:40:22.007Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

577dd46cb8-6sbgm

2022-08-18T19:40:29.782Z

✓ Cube.js server (0.30.46) is listening on 4000

577dd46cb8-6sbgm

2022-08-18T19:40:44.091Z

Q {"message":"Internal Server Error on readiness probe","error":{"error: datab

5877cd774b-wcjmr

2022-08-18T19:40:51.748Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

577dd46cb8-6sbgm

2022-08-18T19:41:00.320Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

577dd46cb8-6sbgm

2022-08-18T19:41:13.035Z

Q {"message":"Internal Server Error on readiness probe","error":{"Error: Called

5877cd774b-wcjmr

2022-08-18T19:41:21.790Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

577dd46cb8-6sbgm

2022-08-18T19:41:29.913Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

5877cd774b-wcjmr

2022-08-18T19:41:31.115Z

Q {"message":"Internal Server Error","error":{"Error: Compile errors:\nSyntax

8476cbc9d5-9t558

2022-08-18T19:41:45.383Z

✓ Cube.js server (0.30.46) is listening on 4000

5877cd774b-wcjmr

2022-08-18T19:41:51.763Z

Q {"message":"Refresh Scheduler Error","error":{"Error: Compile errors:\nSyntax

cube.dev

57

Cube API logs when deploying new builds

The screenshot displays the Cube Cloud web interface for a deployment named 'adnan-test-4'. The interface includes a sidebar with navigation options: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The main content area is titled 'Overview' and shows the deployment is on 'us-central1 (Iowa)'. It features sections for REST API, SQL API, and GraphQL API, each with a 'How to connect your application' link and a URL. The REST API section has a 'Deploy SQL API' button. The SQL API section explains that Cube can function as a headless BI layer. The GraphQL API section is marked as 'beta'. Below these is a 'Resources & Logs' section with a 'Manage resources' link, listing 'Single instance' and 'Build #1934' with links to 'Cube API' and 'Cube Store'. A note at the bottom states: 'Single instance deployment is designed for development use cases. Consider switching to cluster type for production workload.' On the right, an 'Activity Log' section shows a list of builds with their status, user, and timestamp. A warning message at the top right states: 'Cannot deploy latest build. Please make sure your Cube project is configured correctly. Check logs Upload updated Cube project'. At the bottom right, there is a link to 'Connect to GitHub'.

cube cloud Deployments / adnan-test-4 VPCs Docs AD

master Enter Development Mode

Overview

Playground

Schema

Queries

Pre-Aggregations

Metrics

Settings

Overview

us-central1 (Iowa)

REST API

How to connect your application

<https://apricot-koala.gcp-us-central1.cubecloudapp.dev/cubejs->

SQL API

With this new API, Cube can now function as a headless BI layer to provide metrics to any SQL-compatible querying and visualization tool.

Deploy SQL API

GraphQL API

beta

How to connect your application

<https://apricot-koala.gcp-us-central1.cubecloudapp.dev/cubejs->

Resources & Logs

Manage resources

Single instance	Build #1934
✓ Cube API	
✓ Cube Store	

Single instance deployment is designed for development use cases. Consider switching to cluster type for production workload.

Activity Log

View all

✓ Build #1935	adnan@cube.dev	Aug 18 at 21:41
75af601c: removing breaking change		
✓ Env var update		Aug 18 at 21:40
✓ Build #1934	adnan@cube.dev	Aug 18 at 21:33
f491475b: a breaking change		
✓ Build #1933	adnan@cube.dev	Aug 18 at 12:17
a4555386: Merge develop to master		

You can connect your Git repository to Cube Cloud

[Connect to GitHub](#)

Cannot deploy latest build.
Please make sure your Cube project is configured correctly.
[Check logs](#) [Upload updated Cube project](#)

Health checks

- Build failed but deployment still works
- Health checks
 - <https://cube.dev/docs/deployment/production-checklist#set-up-health-checks>
- Fallback to existing deployment
- Additional reliability

Demo 3

Running a build with a broken config

Part 2.2

Query tracing and drill-downs

Query tracing and drill-downs

API requests + Database SQL queries

- logs
- traces
- drill-downs

API request logs

- Overview with metrics
- Query history for the last 24 hours
- Filter by duration, cache, pre-aggregation, and errors

API request logs

The screenshot displays the 'cube cloud' web interface. The top navigation bar includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below the navigation bar, there's a section with 'master' and 'Enter Development Mode' buttons, along with a status indicator 'API is ready (Build #1926)' and a 'Copy API URL' button.

The main content area is titled 'Queries' and has tabs for 'API' (selected) and 'Database'. A filter dropdown shows 'from 24 hours ago to now'. Below the tabs, there are filters for 'Duration: Any', 'Cache Status: All', 'Use Aggregations: Any', and 'Has error: Any'.

The table below lists the API queries with their durations, times, cache statuses, and statuses.

	Duration	Time	Cache	Status
["Orders.count"], "order": {"Orders.count": "desc"}, "dimensions": ["Orde...	754ms	Aug 17 at 13:31:42	No	✓
["Orders.count"], "order": {"Orders.count": "desc"}, "dimensions": ["Orde...	384ms	Aug 17 at 13:31:37	No	✓
["Customer.count"]}	297ms	Aug 17 at 13:31:13	No	✓
["Orders.count"], "timeDimensions": [{"dimension": "Orders.oOrderdate...	208ms	Aug 17 at 13:30:59	-	✓ ⚡
["Orders.count"], "timeDimensions": [{"dimension": "Orders.oOrderdate...	22ms	Aug 17 at 13:30:53	-	✓ ⚡
["Orders.count"], "timeDimensions": [{"dimension": "Orders.oOrderdate...	688ms	Aug 17 at 13:29:56	-	✓ ⚡

API request logs

The screenshot shows the Cube Cloud web interface. The top navigation bar includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below the navigation bar, there's a section with 'master' and a button to 'Enter Development Mode'. A status message indicates 'API is ready (Build #1926)' with a 'Copy API URL' button.

The main content area is titled 'Queries' and has tabs for 'API' and 'Database'. A filter dropdown shows 'from 24 hours ago to now'. Below the tabs, there are filters for 'Duration: > 1 second', 'Cache Status: All', 'Use Aggregations: Any', and 'Has error: Any'.

A table displays the query logs:

	Duration	Time	Cache	Status
s":["Lineitem.count"],"timeDimensions":[{"dimension":"Lineitem.IReceipt...	22.0s	Aug 17 at 13:49:14	No	✓

API request drill down

- JSON query
- SQL query
- If a / which pre-aggregation was used
- Tracing with spans

API request drill down

The screenshot displays the Cube Cloud web interface. At the top, the header shows 'cube cloud' and 'Deployments / adnan-test-4'. On the right, there are links for 'VPCs', 'Docs', and a user profile 'AD'. Below the header, a navigation bar includes a 'master' dropdown, an 'Enter Development Mode' button, and a status indicator 'API is ready (Build #1926)' with a 'Copy API URL' link.

The left sidebar contains a menu with the following items: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The 'Overview' section is currently active.

The main content area is titled 'Request' with a unique ID '664b9db7-d6a0-4024-9251-bff726687b02'. A 'Run in Playground' button is located in the top right corner of this section. Below the title, there are four summary cards:

- Duration:** 22.0s
- Type:** Query
- Client:** Cube Playground
- Cache Status:** No

Below these cards, there are tabs for 'Query', 'SQL', 'Pre-Aggregation', 'Queue', and 'Request Span'. The 'Query' tab is selected, showing a JSON query definition:

```
{
  "measures": [
    "Lineitem.count"
  ],
  "timeDimensions": [
    {
      "dimension": "Lineitem.lReceiptdate",
      "granularity": "day"
    }
  ],
  "order": {
    "Lineitem.lReceiptdate": "asc"
  }
}
```

A back arrow button is visible at the bottom left of the main content area.

API request drill down

The screenshot shows the cube.dev API interface. At the top, there's a status bar with 'master' selected, a 'Enter Development Mode' button, and a green checkmark indicating 'API is ready (Build #1926)' with a 'Copy API URL' link. The left sidebar contains navigation links: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The main content area is titled 'Request' with a back arrow and a request ID '664b9db7-d6a0-4024-9251-bff726687b02'. A 'Run in Playground' button is in the top right. Below the title are four summary cards: Duration (22.0s), Type (Query), Client (Cube Playground), and Cache Status (No). Under these cards are tabs for Query, SQL, Pre-Aggregation, Queue, and Request Span. The 'SQL' tab is active, showing a message 'This request triggered the following query to your database' and a query ID 'b96bf5577a3cf6f818380623a39bcc79' with a duration of '10.9s'. The SQL query is displayed in a code block with syntax highlighting:

```
SELECT
  date_trunc(
    'day',
    (
      "lineitem".l_receiptdate :: timestampz AT TIME ZONE 'UTC'
    )
  ) "lineitem__l_receiptdate_day",
  count(*) "lineitem__count"
FROM
  public.lineitem AS "lineitem"
GROUP BY
  1
ORDER BY
  1 ASC
LIMIT
  10000
```

API request drill down

The screenshot displays the Cube Cloud web interface. At the top, the header includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'. Below the header, a sidebar on the left contains navigation items: 'Overview', 'Playground', 'Schema', 'Queries', 'Pre-Aggregations', 'Metrics', and 'Settings'. The main content area is titled 'Request' with a back arrow and a request ID '7bf8fccb-1a31-40f8-a4d9-89d5021def38'. A 'Run in Playground' button is located in the top right of the main area. Below the title, four cards display key metrics: 'Duration' (4.8s), 'Type' (Query), 'Client' (Cube Playground), and 'Cache Status' (No). At the bottom, a tabbed interface shows 'Query', 'SQL', 'Pre-Aggregation' (selected), 'Queue', and 'Request Span'. The 'Pre-Aggregation' tab contains a message: 'Query was not accelerated with pre-aggregation. [Accelerate](#)'. A back arrow is visible in the bottom left corner of the main content area.

cube cloud Deployments / adnan-test-4 VPCs Docs AD

master </> Enter Development Mode API is ready (Build #1933) Copy API URL

← Request 7bf8fccb-1a31-40f8-a4d9-89d5021def38 Run in Playground

Duration	Type	Client	Cache Status
4.8s	Query	Cube Playground	No

Query SQL ▲ Pre-Aggregation Queue Request Span

Query was not accelerated with pre-aggregation. [Accelerate](#)

API request drill down

The screenshot shows the 'cube cloud' interface. At the top, it says 'Deployments / adnan-test-3'. There's a button to 'Enter Development Mode' and a status 'API is ready (Build #1974)'. A sidebar on the left contains links to Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings. The main area is titled 'Request' with the ID 'c2e232e7-751e-4814-bd4f-e5b9cf8d7f36'. Below this, four cards display key metrics: Duration (2.2s), Type (Query), Client (Cube Playground), and Cache Status (No). A 'Run in Playground' button is in the top right. Below the cards, a tabbed interface shows 'Request Span' selected. The table below lists events in the request span.

Event	Timestamp	Time Since Start Duration	Sql Query Key Md5	Request Id	Waiting for Request Id	Details
Incoming network usage	Aug 24 at 10:44:32	-1ms		c2e232e7-751e-4814-bd4f-e5b9cf8d7f36-span-1		{"service":"api-http","t
Load Request	Aug 24 at 10:44:32	0ms		c2e232e7-751e-4814-bd4f-e5b9cf8d7f36-span-1		{"securityContext":{"ex

Tracing for SQL queries

- Overview of all SQL queries that were executed
- Query history for the last 24 hours
- Filter by duration, and errors

Tracing for SQL queries

The screenshot displays the cube.dev web application interface. On the left is a sidebar with navigation links: Overview, Playground, Schema, Queries (selected), Pre-Aggregations, Metrics, and Settings. The main content area is titled 'Queries' and has tabs for 'API' and 'Database'. A search bar contains the text 'contains'. Below the search bar are filters for 'Duration: Any' and 'Has error: Any'. A table lists the following queries and their execution details:

Query	Duration	Time	Status
SELECT date_trunc('day', ("orders".o_orderd...	6.7s	Aug 17 at 16:00:04	✓
SELECT date_trunc('day', ("orders".o_orderd...	6.7s	Aug 17 at 15:00:03	✓
SELECT date_trunc('day', ("orders".o_orderd...	7.3s	Aug 17 at 14:00:03	✓
SELECT date_trunc('day', ("lineitem".l_receip...	10.9s	Aug 17 at 13:49:36	✓
SELECT date_trunc('day', ("lineitem".l_receip...	21.6s	Aug 17 at 13:49:14	✓
SELECT "orders".o_orderstatus "orders__o_o...	569ms	Aug 17 at 13:31:42	✓
SELECT "orders".o_orderstatus "orders__o_o...	373ms	Aug 17 at 13:31:37	✓
SELECT count(*) "customer__count" FROM p...	286ms	Aug 17 at 13:31:13	✓
SELECT date_trunc('day', ("orders".o_orderd...	7.0s	Aug 17 at 13:03:49	✓

At the top right of the interface, it indicates 'API is ready (Build #1926)' and provides a 'Copy API URL' button. A dropdown menu in the top right corner shows 'from 24 hours ago to now'.

SQL query drill down

- Details about the SQL query
- Trace of events that occurred

SQL query drill down

The screenshot displays the Cube Cloud web interface. At the top, the header shows 'cube cloud' and 'Deployments / adnan-test-4'. On the right, there are links for 'VPCs', 'Docs', and a user profile 'AD'. Below the header, a navigation bar includes a dropdown for 'master' and a button for '</> Enter Development Mode'. On the right side of this bar, it indicates 'API is ready (Build #1926)' and a 'Copy API URL' button.

The left sidebar contains a list of navigation items: Overview, Playground, Schema, Queries (highlighted), Pre-Aggregations, Metrics, and Settings.

The main content area is titled 'SQL Query' with a unique identifier 'b96bf5577a3cf6f818380623a39bcc79'. Below the title, three summary cards are displayed: 'Total Duration' with a value of '21.6s', 'GB processed' with a value of '0', and 'Queue' with the name 'SQL_QUERY_STANDALONE_default'.

Below the summary cards, there are two tabs: 'Query' (selected) and 'Events'. The 'Query' tab shows a SQL query in a code editor:

```
SELECT
  date_trunc(
    'day',
    (
      "lineitem".l_receiptdate :: timestampz AT TIME ZONE 'UTC'
    )
  ) "lineitem__l_receiptdate_day",
  count(*) "lineitem__count"
FROM
  public.lineitem AS "lineitem"
GROUP BY
  1
ORDER BY
  1 ASC
LIMIT
  10000
```

SQL query drill down

The screenshot displays the Cube Cloud web interface. At the top, the header includes the 'cube cloud' logo, the deployment name 'adnan-test-3', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'. A secondary bar contains a button to 'Enter Development Mode' and a status indicator 'API is ready (Build #1974)' with a 'Copy API URL' link.

The left sidebar contains a menu with the following items: Overview, Playground, Schema, Queries (highlighted), Pre-Aggregations, Metrics, and Settings.

The main content area is titled 'SQL Query' with a unique ID '9136adb8a4d8456e8c7cf2efca6b5826'. Below the title, three summary cards are shown: 'Total Duration' at 1.3s, 'GB processed' at 0, and 'Queue' set to 'SQL_QUERY_STANDALONE_default'.

Below the summary cards, there are two tabs: 'Query' and 'Events'. The 'Events' tab is active, displaying a table of query execution events.

Event	Timestamp	Time Since Start Duration	Sql Query Key Md5	Request Id	Details
Added to queue	Aug 24 at 10:44:32	656ms	9136adb8	c2e232e7-751e-4814-bd4f-e5b9cf8d7f36-span-1	{"priority":0,"queueSize":1,"queuePrefix":"S
Performing query	Aug 24 at 10:44:32	684ms	9136adb8	c2e232e7-751e-4814-bd4f-e5b9cf8d7f36-span-1	{"processingId":"1","queueSize":1,"queueP
Waiting for				c2e232e7-751e-4814-	

SQL query drill down

The screenshot shows the Cube Cloud web interface. The top navigation bar includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below this, there's a 'master' dropdown and a button to 'Enter Development Mode'. A status bar indicates 'API is ready (Build #1926)' and provides a 'Copy API URL' link.

The left sidebar contains a menu with the following items: Overview, Playground, Schema, Queries (highlighted), Pre-Aggregations, Metrics, and Settings.

The main content area is titled 'Queries' and has tabs for 'API' and 'Database'. A filter dropdown shows 'from 24 hours ago to now'. Below the tabs, there's a search bar with the text 'contains' and two filter dropdowns: 'Duration: Any' and 'Has error: Any'. A 'Query Key: b96bf5577a3c...' is also displayed.

The table below lists several queries with their durations and execution times:

Query	Duration	Time	Status
SELECT date_trunc('day', ("lineitem"."l_receip...	3.8s	Aug 17 at 16:53:23	✓
SELECT date_trunc('day', ("lineitem"."l_receip...	3.9s	Aug 17 at 16:52:52	✓
SELECT date_trunc('day', ("lineitem"."l_receip...	5.9s	Aug 17 at 16:52:16	✓
SELECT date_trunc('day', ("lineitem"."l_receip...	10.9s	Aug 17 at 13:49:36	✓
SELECT date_trunc('day', ("lineitem"."l_receip...	21.6s	Aug 17 at 13:49:14	✓

Part 2.3

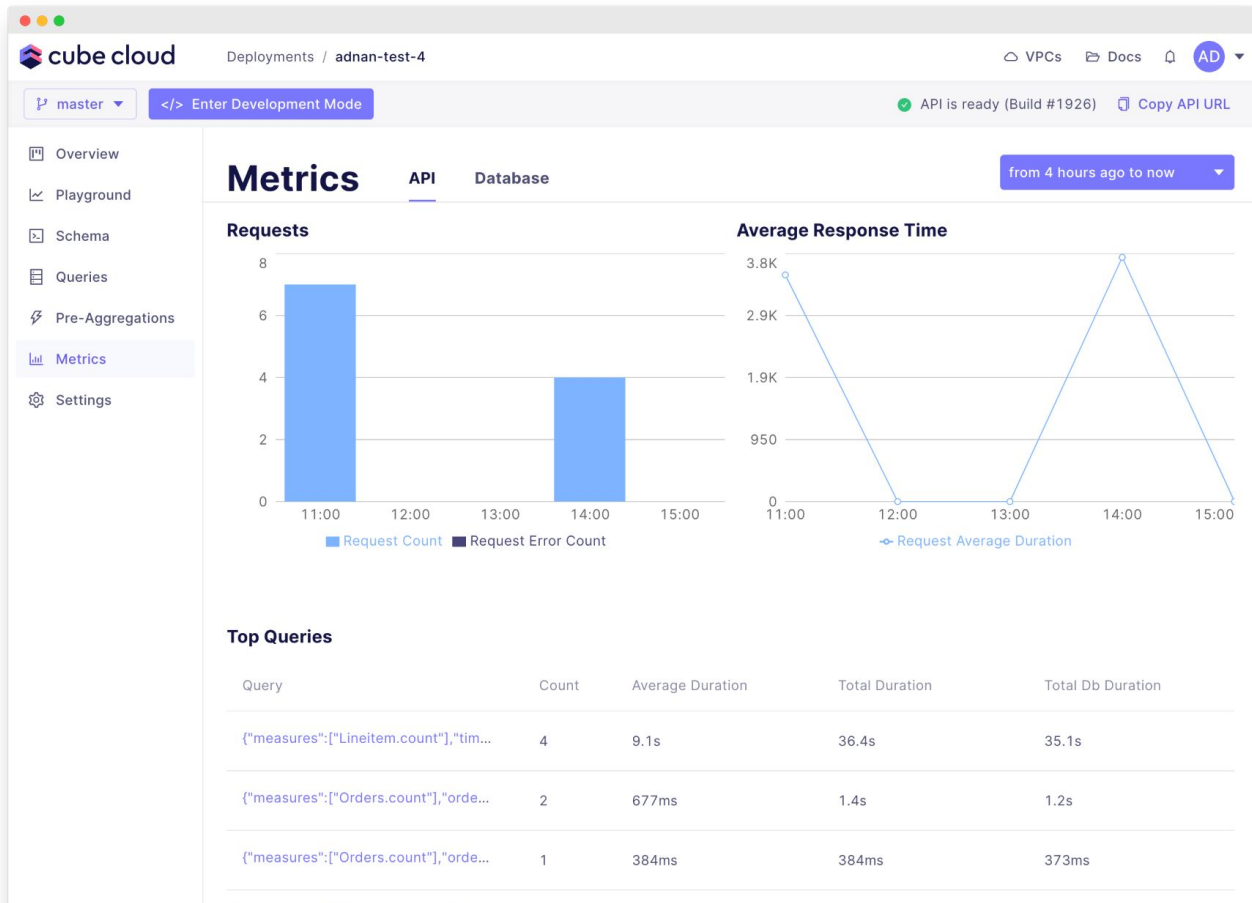
Monitoring metrics

Monitoring metrics

API requests + Database SQL queries

- Top-down overview
- Top requests and queries
- Drill down to specific requests and queries

API request metrics and response times



API request metrics and response times

The screenshot displays the Cube Cloud interface for a deployment named 'adnan-test-4'. The left sidebar contains navigation links: Overview, Playground, Schema, Queries (selected), Pre-Aggregations, Metrics, and Settings. The main content area is titled 'Queries' and shows a table of API request metrics. The table has columns for Query, Duration, Time, and Cache. The 'Duration' column includes a visual bar chart representing the response time. The 'Time' column shows the timestamp of the request. The 'Cache' column indicates whether the request was cached. The table lists four queries, all with a duration of 3.9s, 4.0s, 6.5s, and 22.0s respectively. The 'Time' column shows timestamps from August 17, 2023, at 16:53:23 to 13:49:14. The 'Cache' column shows 'No' for all queries. The interface also includes a top navigation bar with 'cube cloud' logo, 'Deployments / adnan-test-4', and a user profile 'AD'. A status bar at the top right indicates 'API is ready (Build #1926)' and provides a 'Copy API URL' button. A filter bar at the top of the table allows filtering by Duration, Cache Status, Use Aggregations, and Has error, with a query filter set to 'Query: { "measures": ["Lin..." }'.

cube cloud Deployments / adnan-test-4 VPCs Docs AD

master Enter Development Mode API is ready (Build #1926) Copy API URL

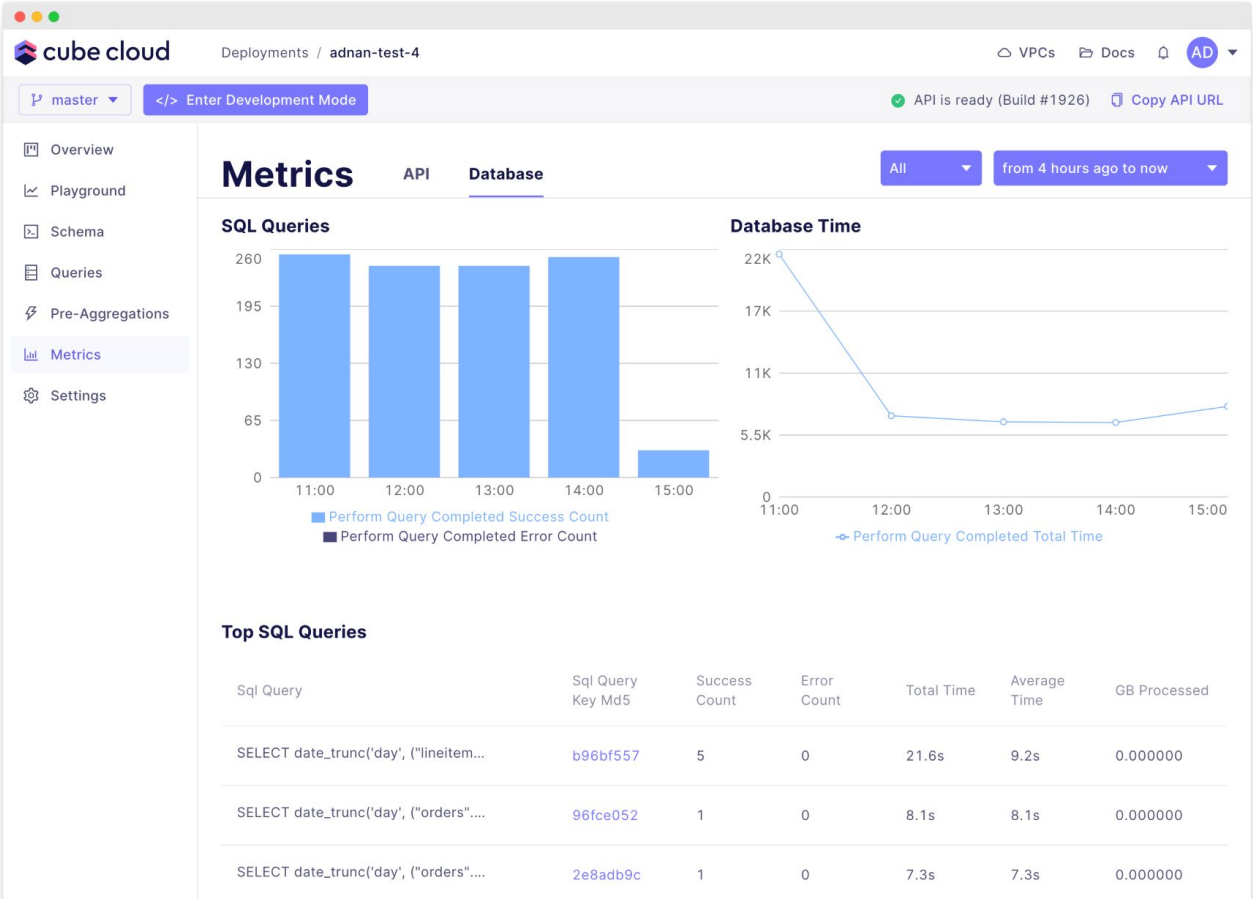
Overview Playground Schema Queries Pre-Aggregations Metrics Settings

Queries API Database from 4 hours ago to now

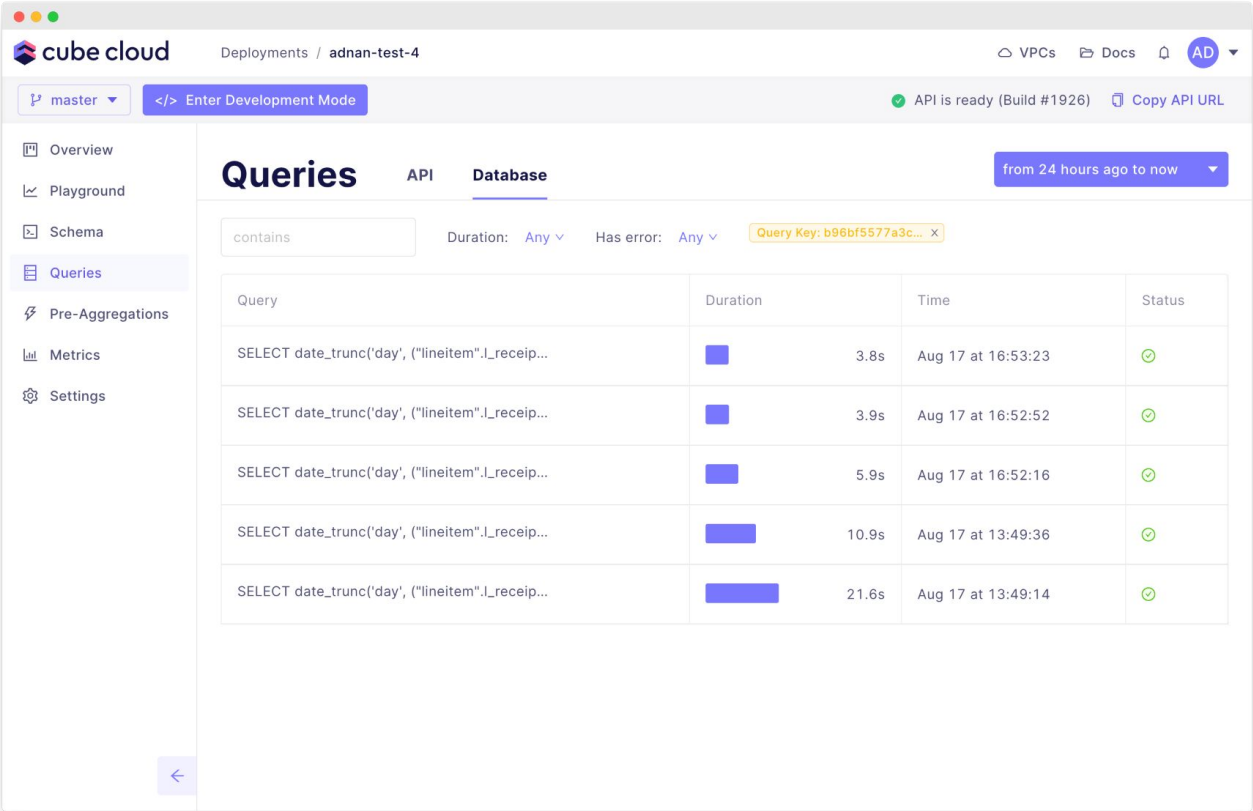
Duration: Any Cache Status: All Use Aggregations: Any Has error: Any Query: { "measures": ["Lin..." } X

Query	Duration	Time	Cache
{"measures":["Lineitem.count"],"timeDimensions":[{"dimension":"Lineitem.IReceipt..."}	3.9s	Aug 17 at 16:53:23	No
{"measures":["Lineitem.count"],"timeDimensions":[{"dimension":"Lineitem.IReceipt..."}	4.0s	Aug 17 at 16:52:52	No
{"measures":["Lineitem.count"],"timeDimensions":[{"dimension":"Lineitem.IReceipt..."}	6.5s	Aug 17 at 16:52:16	No
{"measures":["Lineitem.count"],"timeDimensions":[{"dimension":"Lineitem.IReceipt..."}	22.0s	Aug 17 at 13:49:14	No

Database metrics and response times



Database metrics and response times



API vs. Database tabs

- **API** — only requests against your Cube API endpoints
- **Database** — SQL queries triggered by the API and scheduler
 - **Scheduler** — SQL queries the Cube refresh worker triggers periodically based on the refresh key for your configured pre-aggregations

API vs. Database tabs

- In production an API request $\neq$ a database SQL query
- Pre-aggregation response would not trigger database SQL query
- Rollup-only mode
 - Database tab would only show pre-aggregation refresh queries

On the Observability roadmap

- New APM engine that shows more than 24 hours of data by end of Q3
- Log export will be available to enterprise customers in Q4

Demo 4

Inspecting and accelerating slow queries

Part 2.4

Alerting

Alerting

- Get notified **when issues happen**
- Alert conditions are checked **once per minute**
- If an alert condition is triggered, an email notification will be sent
- An outage email is sent **once** per configured alert
- When the issue is resolved, a resolution email is sent **once**

Alerting

- Alerts are **not available on the free plan**
- Only **admin** users can configure alerts in Cube Cloud

Alert Types

- API outages
- Database timeouts
- Pre-aggregation build failures
- Deployment builds

Deployments and Recipients

- All or specific deployments
- All or specific users

Alerting

The screenshot shows the 'Add alert' dialog box in the 'cube cloud' application. The dialog is centered over a blurred background of the application's main interface, which includes a sidebar with 'Alerts' and a top navigation bar with 'VPCs', 'Docs', and a user profile 'AD'. The dialog itself has a title 'Add alert' and three main sections: 'Event type', 'For deployments', and 'Send alerts to'. Each section contains a list of options with radio or checkbox selectors. The 'Event type' section has five options, with 'All' selected. The 'For deployments' section has two radio options, 'All' and 'Specific', with 'Specific' selected; below it is a list of deployment names, with 'adnan-test-4' checked. The 'Send alerts to' section has two radio options, 'All users on this account' and 'Specific users', with 'Specific users' selected; below it is a list of email addresses, with 'adnan@cube.dev' checked. At the bottom of the dialog, there is a partially visible 'Also send to' section with an input field.

Add alert

Event type

- ☐ API outages
- ☐ Database response timeouts
- ☐ Pre-aggregation build failures
- ☐ Build completed
- ☒ All

For deployments

☐ All ☒ Specific

- ☐ databricks-test
- ☒ adnan-test-4
- ☐ adnan-test-3
- ☐ adnan-test-2
- ☐ test-paco-supabase

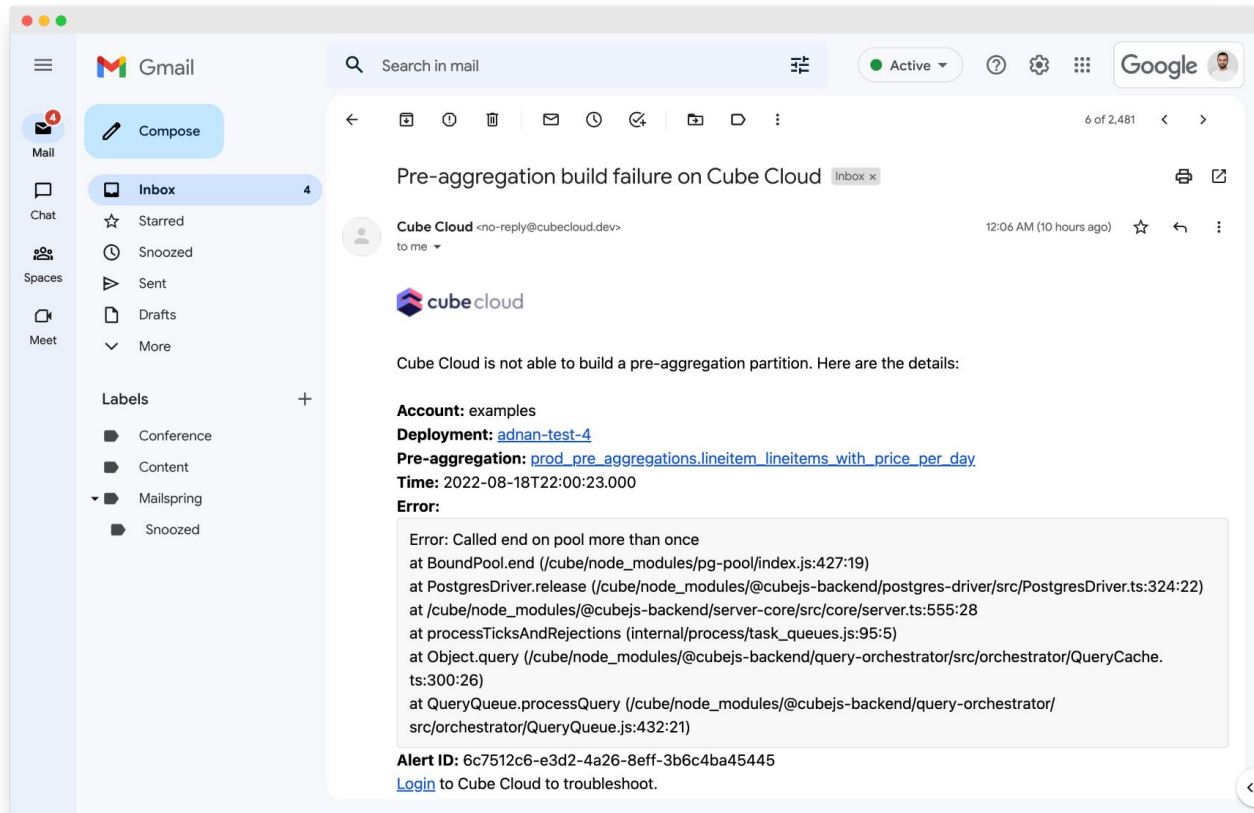
Send alerts to

☐ All users on this account ☒ Specific users

- ☐ artyom@cube.dev
- ☐ evgenii@cube.dev
- ☒ adnan@cube.dev
- ☐ shadid@cube.dev
- ☐ bold.car1172@fastmail.com

Also send to

Alerting



Demo 5

Alerting

Part 3

Performance

Performance

- Pre-aggregations
- Rollup-only mode
- Cluster mode
- Cube Store

Part 3.1

Pre-aggregations

Previous workshops about pre-aggregations!

Advanced Pre-aggregations in Cube (March 30, 2022)

Agenda: [00:00:00 - 00:06:20] Introductions [00:06:21 - 00:19:29] Recap from the previous workshop + recent improvements in pre-aggregations

 https://www.youtube.com/watch?v=JxedV_zl7W4

Advanced Pre-aggregations Workshop



Mastering Cube Pre-Aggregations Workshop

Recording from the Cube Pre-Aggregations Workshop on August 18, 2021
Agenda: [00:00:00 - 00:06:30] Introductions [00:06:31 - 00:20:29] Why

 https://www.youtube.com/watch?v=1_BP3B_wbec&t=0s

August 18, 2021

Pre-aggregations Workshop



 Pavel Tiunov  Rvan Pei

<https://cube.dev/events/pre-aggregations>
<https://cube.dev/events/adv-pre-aggregations>

How do pre-aggregations work?

Raw Data

Order ID	Order Time	Product	Quantity
1	July 20, 2021 10:00 AM	Sugar Cubes	2
2	July 22, 2021 08:00 AM	Ice Cubes	4
3	July 26, 2021 03:00 PM	Ice Cubes	2
4	July 28, 2021 12:00 AM	Sugar Cubes	1
5	August 3, 2021 12:00 PM	Ice Cubes	4
6	August 5, 2021 09:00 AM	Ice Cubes	1

Row Count

~1M (5 years of data)

⚡ Pre-Aggregated Data

Order Time	Product	Quantity
July	Ice Cubes	6
July	Sugar Cubes	3
August	Ice Cubes	5

Row Count

~120

Partitioning

- Use partitions if it takes too long to build your pre-aggregation
- `partitionGranularity` – separate tables for each partition of data
 - `day`, `month`, `quarter`, etc...

Without partitioning

The screenshot shows the Cube Cloud web interface. The top navigation bar includes the 'cube cloud' logo, the current deployment path 'Deployments / intro-to-cube-cloud-slow-pre-aggregation-build', and links for 'VPCs', 'Docs', and a user profile 'AD'. A status bar indicates 'API is ready (Build #53)' and provides a 'Copy API URL' button. The left sidebar contains navigation links: Overview, Playground, Schema, Queries, Pre-Aggregations (highlighted), Metrics, and Settings. The main content area is titled 'Lineitem' with the identifier 'lineitemsWithPricePerDay'. Under the 'Refresh' section, it shows 'Every 1 hour', 'No incremental updates', and 'Automated refreshes'. Below this is a tabbed interface with 'Partitions', 'Definition', 'Preview', 'Used By', and 'Indexes'. The 'Partitions' tab is active, displaying a table with columns for 'Last started at', 'Duration', and 'Partition size'. A single partition is listed with a green status icon, the name 'Lineitem.lineitemsWithPricePerDay', a start time of 'Aug 31 at 12:00:21', a duration of '8.7m', and a size of '27.59MB'. A 'Build All' button is located in the top right of the table area.

cube cloud Deployments / intro-to-cube-cloud-slow-pre-aggregation-build VPCs Docs AD

</> Enter Development Mode API is ready (Build #53) Copy API URL

Overview Playground Schema Queries **Pre-Aggregations** Metrics Settings

← **Lineitem** lineitemsWithPricePerDay

Refresh

- ⌚ Every 1 hour
- ✗ No incremental updates
- ✓ Automated refreshes

Partitions Definition Preview Used By Indexes Build All

	Last started at	Duration	Partition size
✓ Lineitem.lineitemsWithPricePerDay	Aug 31 at 12:00:21	8.7m	27.59MB

With partitioning

The screenshot displays the Cube Cloud web interface for a deployment named 'adnan-test-3'. The left sidebar contains navigation links: Overview, Playground, Schema, Queries, Pre-Aggregations (highlighted), Metrics, and Settings. The main content area shows the configuration for the 'Lineitem' table, specifically the 'lineitemsWithPricePerDay' view. A 'Refresh' section indicates the table is updated every 1 hour with incremental refreshes over a 7-day window. Below this, the 'Partitions' tab is active, showing a table of partitions for the build range 'Jan 1, 1992 - Jan 1, 1999'. The table lists partitions by month from July 1998 to December 1998, each with a status icon, last started time, duration, and size. A 'Build All' button is located at the top right of the partitions table.

cube cloud Deployments / adnan-test-3

</> Enter Development Mode API is ready (Build #1958) Copy API URL

← Lineitem lineitemsWithPricePerDay

Refresh

- ⌚ Every 1 hour
- ✓ Incremental refresh update window: last 7 days
- ✓ Automated refreshes

Partitions Definition Preview Used By Indexes Build All

Build range: Jan 1, 1992 - Jan 1, 1999

☐		Last started at	Duration	Partition size
☐	December 1998	Aug 19 at 11:22:14	12.3s	21.66KB
☐	November 1998	Aug 19 at 11:22:44	13.7s	107.45KB
☐	October 1998	Aug 19 at 11:22:44	10.1s	207.00KB
☐	September 1998	Aug 19 at 11:23:14	22.4s	290.71KB
☐	August 1998	Aug 19 at 11:23:14	18.0s	368.37KB
☐	July 1998	Aug 19 at 11:23:44	18.1s	377.43KB

Pre-aggregation preview

cube cloud Deployments / intro-to-cube-cloud-partitions

</> Enter Development Mode

API is ready (Build #52) Copy API URL

Overview
Playground
Schema
Queries
Pre-Aggregations
Metrics
Settings

← Lineitem

lineitemsWithPricePerDay

Refresh

- Every 1 hour
- ✓ Incremental refresh update window: last 7 days
- ✓ Automated refreshes

Partitions Definition **Preview** Used By Indexes

lineitem__l_extendedprice	lineitem__l_receiptdate_day	lineitem__count
914	1998-12-02T00:00:00.000Z	1
973.06	1998-12-06T00:00:00.000Z	1
984	1998-12-01T00:00:00.000Z	1
1005.05	1998-12-01T00:00:00.000Z	1
1023.02	1998-12-07T00:00:00.000Z	1
1033.03	1998-12-09T00:00:00.000Z	1

Pre-aggregation build history

Deployments / intro-to-cube-cloud-partitions

VPCs

Docs

AD

</> Enter Development Mode

API is ready (Build #52)

Copy API URL

Overview

Playground

Schema

Queries

Pre-Aggregations

Metrics

Settings

Pre-Aggregations

Pre-Aggregations

Build History

from 24 hours ago to now

		Trigger	Added to queue at	Started
success	Lineitem.lineitemsWithPricePerDay / January 1992	Refresh Worker	Aug 31 at 12:01:38	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / March 1992	Refresh Worker	Aug 31 at 12:01:08	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / February 1992	Refresh Worker	Aug 31 at 12:01:08	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / May 1992	Refresh Worker	Aug 31 at 12:00:38	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / April 1992	Refresh Worker	Aug 31 at 12:00:38	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / June 1992	Refresh Worker	Aug 31 at 12:00:08	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / July 1992	Refresh Worker	Aug 31 at 12:00:08	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / September 1992	Refresh Worker	Aug 31 at 11:59:38	Aug 31 at 1
success	Lineitem.lineitemsWithPricePerDay / August 1992	Refresh Worker	Aug 31 at 11:59:38	Aug 31 at 1

Part 3.2

Scaling with

Cluster mode & Rollup-only mode

Cube Cloud deployment types

- Single instance vs Cluster mode
- Determine features, scalability, and availability of your deployment

Single Instance Deployments

- For development use-cases only
- Hibernated if not used for over an hour
- No dedicated Refresh Worker

Cluster Deployments

- Designed for production workloads
- Autoscaling
- Pre-aggregation warmup
- Rollup-only mode

Cluster Deployments

Cluster deployment includes the following components:

- Cube API instances
- Cube Store cluster
- Cube refresh workers
- Redis for Cache and Queue — deprecated, will be replaced by Cube Store by end of Q4

Cluster Mode

The screenshot shows the Cube Cloud web interface. At the top, the header includes the 'cube cloud' logo, the deployment path 'Deployments / adnan-test-3', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'. A purple button labeled '</> Enter Development Mode' is on the left, and a status bar on the right indicates 'API is ready (Build #1958)' with a 'Copy API URL' link.

The left sidebar contains a menu with the following items: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings (which is highlighted).

The main content area is titled 'Settings' and has three tabs: 'General' (selected), 'Configuration', and 'Build & Deploy'.

The 'General' tab is divided into two sections:

- Deployment**:
 - Type**: Radio buttons for 'Single Instance' and 'Cluster' (selected).
 - Name**: A text input field containing 'adnan-test-3'.
 - Cube version**: A dropdown menu showing 'Stable (0.30.46)'.
 - Cloud provider**: A dropdown menu showing 'Google Cloud Platform'.
 - Cloud region**: A text input field containing 'gcp-us-central1-c-3'.
 - Cube API**: A text input field containing 'https://electric-fowl.gcp-us-central1.cubecloudapp.dev/cube/' with a copy icon.
 - A red 'Delete Deployment' button is at the bottom.
- Custom domain**:
 - A text input field for a custom domain and an 'Add' button.
 - Below the input field, a note states: 'Before adding a custom domain here, set a CNAME record for the custom domain mapped to **electric-fowl.gcp-us-central1.cubecloudapp.dev**, and wait for a few minutes for that to take effect. Then add your custom domain here, which will take effect after several minutes. [Contact us](#) if you encounter any problems.'

Cluster Mode

The screenshot shows the Cube Cloud web interface. At the top, the header includes the 'cube cloud' logo, the breadcrumb 'Deployments / intro-to-cube-cloud-partitions', and navigation links for 'VPCs', 'Docs', and a user profile 'AD'. A purple button labeled '</> Enter Development Mode' is on the left. On the right, a status bar shows 'API is ready (Build #52)' with a green checkmark and a 'Copy API URL' button.

The left sidebar contains a menu with icons and labels: Overview, Playground, Schema, Queries, Pre-Aggregations, Metrics, and Settings (which is highlighted).

The main content area is titled 'Settings' and has three tabs: 'General', 'Configuration' (which is active), and 'Build & Deploy'.

Under the 'Configuration' tab, there are two main sections:

- Configuration flags:** This section contains three toggle switches:
 - Rollup Only mode:** Turned on (blue).
 - Enable SQL API:** Turned off (grey).
 - Warm-up pre-aggregations before deploying API:** Turned on (blue).
- Allow queries from these IPs:** A text box containing a list of IP addresses:
35.224.101.64
35.192.213.232
104.154.140.129
35.224.155.76

Below these sections is the **Cube environment variables** section, which includes a link 'Learn more about Cube env vars'. It features a 'Show All Values' button and a table of environment variables:

Variable Name	Value	Copy	Delete
CUBEJS_API_SECRET			
CUBEJS_SQL_USER			
CUBEJS_SQL_PASSWORD			
CUBESQL_REWRITE_ENGINE			
CUBEJS_SCHEDULED_REFRESH_DEFAULT			

Cluster Mode

cube cloud

Deployments / intro-to-cube-cloud-partitions

VPCsDocsAD

</> Enter Development Mode

OverviewPlaygroundSchemaQueriesPre-AggregationsMetricsSettings

Overviewus-central1 (Iowa)

REST APIHow to connect your application

https://silver-meadowlark.gcp-us-central1.cubecloudapp.dev/cut

SQL APIWith this new API, Cube can now function as a headless BI layer to provide metrics to any SQL-compatible querying and visualization tool.

Deploy SQL API

GraphQL APIbetaHow to connect your application

https://silver-meadowlark.gcp-us-central1.cubecloudapp.dev/cut

Resources & LogsManage resources

ClusterBuild #52

Cube APICube APIDisabled

RedisRedis enabled

Cube Refresh WorkersCube Refresh Workers enabled

Cube StoreCube Store enabled

Activity LogView all

Cube Sql API disabledAug 31 at 12:24

Cluster mode enabledAug 31 at 12:23

Build #52Aug 31 at 11:39

Build #51Aug 31 at 11:38

Env var updateAug 31 at 11:38

You can deploy changes to your Cube project via Cube CLI or from your CI/CD.

How to deploy

Demo 6

Performance

Part 3.3

Cube Store

Cube Store in Cube Cloud

- Huge!
- Shared across accounts
- Based on resource consumption

Cube Store in Cube Cloud

- Scale 1 with 1.5 GB of data takes 8 seconds
- Scale 100 with 150 GB of data takes 1.7 minutes

Scale 1 – 1.5 GB dataset

The screenshot shows the Cube Cloud web interface. The top navigation bar includes the 'cube cloud' logo, the deployment name 'adnan-test-4', and links for 'VPCs', 'Docs', and a user profile 'AD'. Below the navigation bar, there's a status bar with 'master' selected, a button to 'Enter Development Mode', and a message 'API is ready (Build #1982)' with a 'Copy API URL' link.

The left sidebar contains a menu with the following items: Overview, Playground, Schema, Queries, Pre-Aggregations (highlighted), Metrics, and Settings.

The main content area is titled 'Pre-Aggregations' and has two tabs: 'Pre-Aggregations' (active) and 'Build History'. It displays a table with the following data:

		Last started at	Duration	Partition size	Partitions
●	Orders ordersByDay	Aug 26 at 16:00:14	7.2s	10.02KB	1 / 1

Scale 100 – 150 GB dataset

The screenshot displays the Cube Cloud web interface. At the top, the header shows 'cube cloud' and the deployment path 'Deployments / kcdc2022-scale100'. On the right, there are links for 'VPCs', 'Docs', and a user profile 'AD'. Below the header, a status bar indicates 'API is ready (Build #1975)' and provides a 'Copy API URL' button. The left sidebar contains a navigation menu with items: Overview, Playground, Schema, Queries, Pre-Aggregations (highlighted), Metrics, and Settings. The main content area is titled 'Pre-Aggregations' and has two sub-tabs: 'Pre-Aggregations' (active) and 'Build History'. A table lists the pre-aggregations:

	Last started at	Duration	Partition size	Partitions
● Orders ordersByDay	Aug 22 at 16:01:50	1.7m	11.50KB	1 / 1

Scale 100 – 150 GB dataset with partitioning

cube cloud Deployments / kcdc2022-scale100

master </> Enter Development Mode API is ready (Build #1976) Copy API URL

Overview Playground Schema Queries Pre-Aggregations Metrics Settings

← Orders ordersByDay

Refresh

- ⌚ Every 1 hour
- ✓ Incremental refresh update window: last 7 days
- ✓ Automated refreshes

Partitions Definition Preview Used By Indexes Build All

Build range: Jan 1, 1992 - Sep 1, 1998

☐	Last started at	Duration	Partition size
☐ ✓ August 1998	Aug 22 at 16:15:38	43.1s	82.00B
☐ ✓ July 1998	Aug 22 at 16:16:08	29.3s	244.00B
☐ ✓ June 1998	Aug 22 at 16:16:38	35.8s	233.00B
☐ ✓ May 1998	Aug 22 at 16:16:38	36.2s	240.00B
☐ ✓ April 1998	Aug 22 at 16:17:38	41.3s	239.00B
☐ ✓ March 1998	Aug 22 at 16:17:38	41.5s	242.00B



Resources

To learn more visit...

- Documentation:
 - IDE and Dev Mode – <https://cube.dev/docs/cloud/dev-tools/cube-ide>
 - Alerts – <https://cube.dev/docs/cloud/dev-tools/alerts>
 - Deploys – <https://cube.dev/docs/cloud/deployments>
 - Tracing – <https://cube.dev/docs/cloud/inspecting-queries>
- Blog posts:
 - <https://cube.dev/blog/cube-cloud-deep-dive-starting-a-new-cube-app>
 - <https://cube.dev/blog/migrating-from-self-hosted-to-cube-cloud>
 - <https://cube.dev/blog/cube-cloud-deep-dive-mastering-pre-aggregations>
 - <https://cube.dev/blog/announcing-cube-cloud>
 - <https://cube.dev/blog/introducing-cloud-alerts>
- Community support:
 - Discourse: <https://forum.cube.dev/>
 - Slack: <http://cube-js.slack.com/>

