



Real-time Demo Workshop

Table of Contents

| | |
|--|----|
| Table of Contents | 2 |
| Workshop User Instructions | 3 |
| Creating AWS Workshop Pre-Requisites | 3 |
| Creating a CrateDB Cluster | 9 |
| Creating AWS Workshop Environment | 11 |
| Installing the demo source code | 14 |
| Running the data producer | 14 |
| AWS Lambda function for Data Consumer | 15 |
| Create a trigger for the AWS Lambda function | 16 |
| Installing Grafana onto AWS EC2 instance | 17 |
| Update packages..... | 17 |
| Add the Grafana repository | 17 |
| Install Grafana | 17 |
| Enable and start the service | 17 |
| Open Grafana port | 17 |
| Import dashboards | 18 |

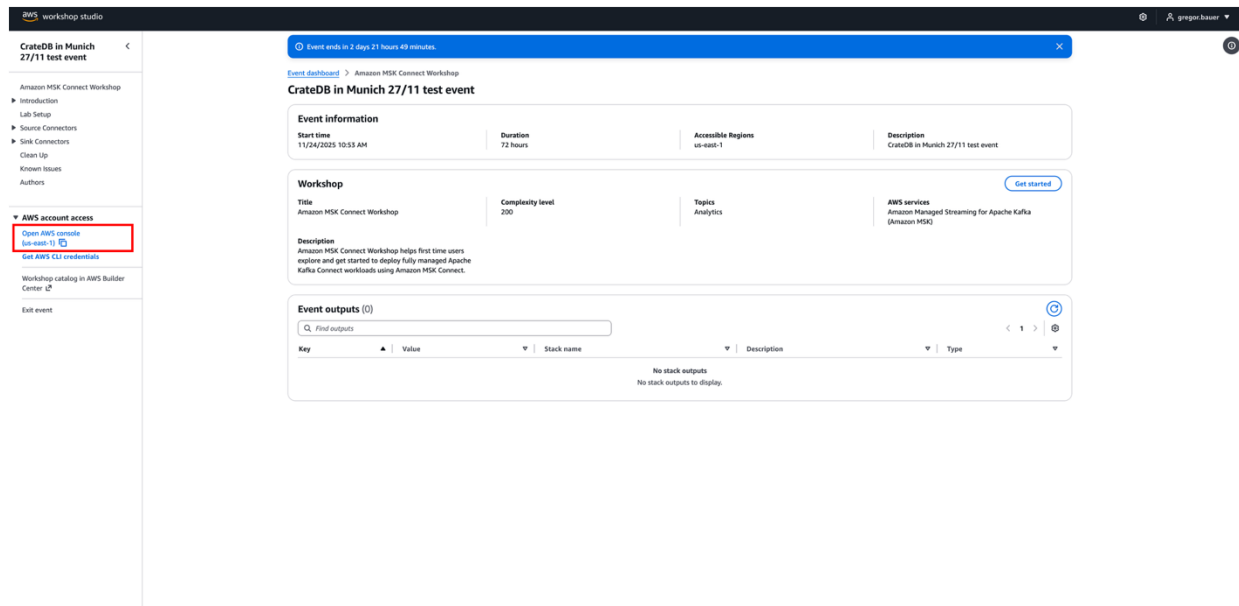
Workshop User Instructions

Creating AWS Workshop Pre-Requisites

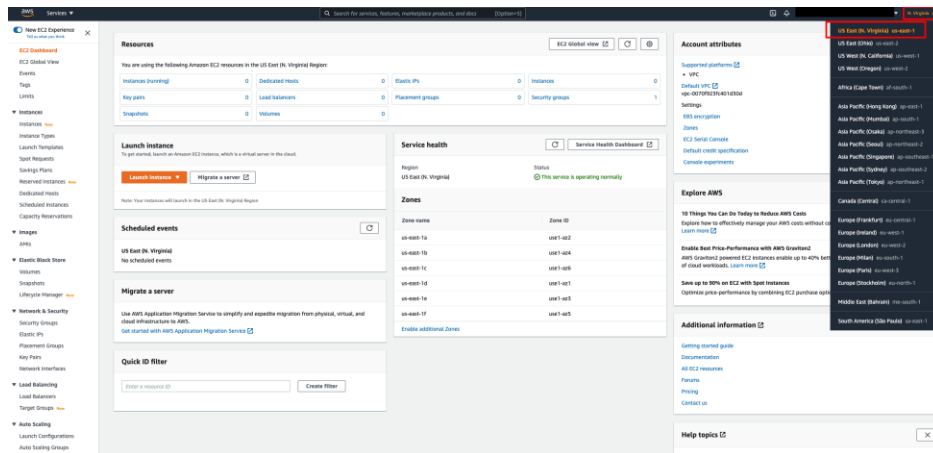
Note: Please use the exact names provided in this guide for naming instances otherwise build steps might fail. Use copy paste to avoid errors.

Note: If you already have AWS console it's best to use another browser

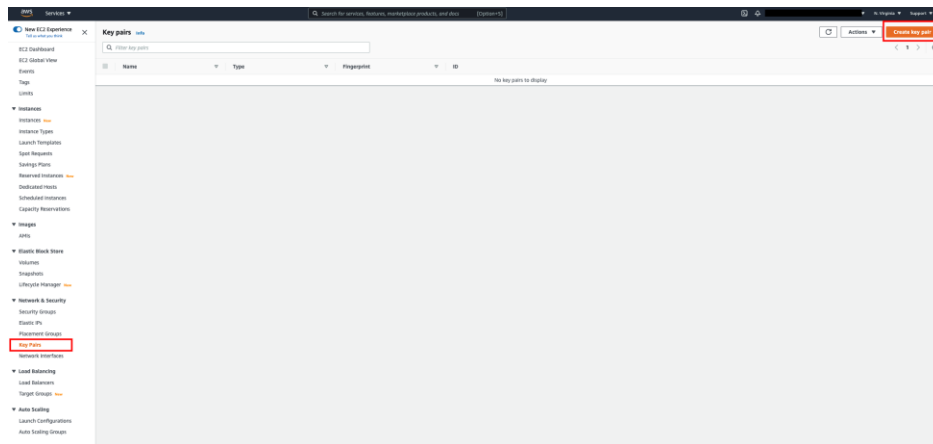
1. Click on the provided link and add the email you registered for the event to get an OTP
2. After entering the OTP you received go to the Amazon Console



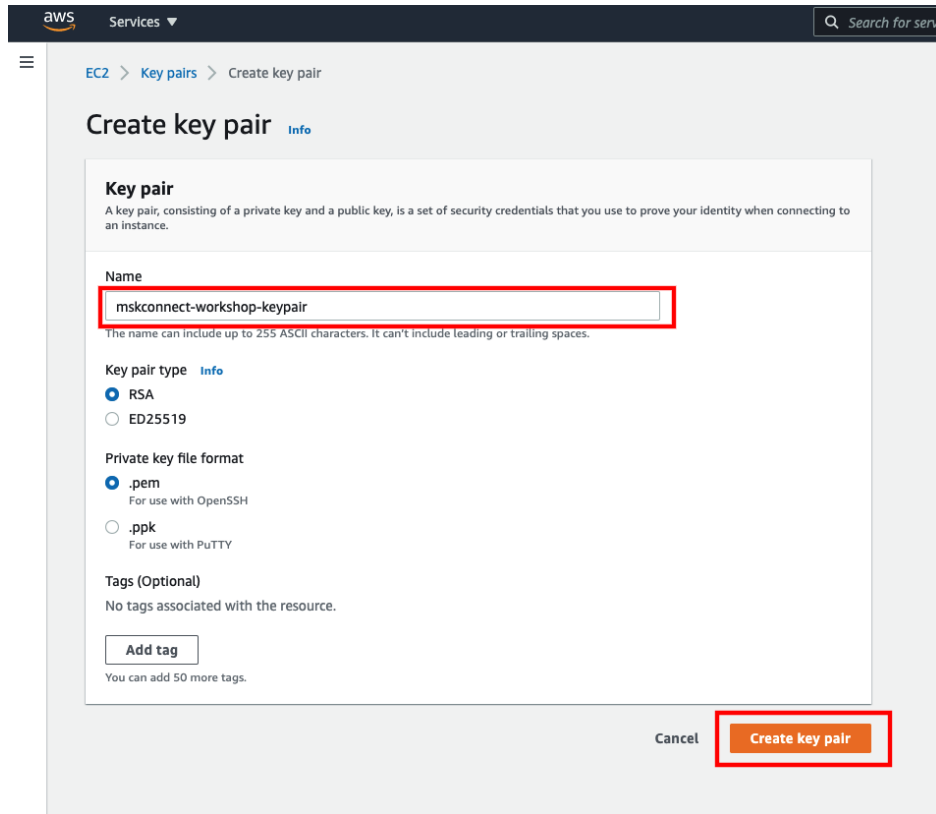
3. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> Choose region **us-east-1** this will be the region that you are using for the workshop.



2. Choose **Key Pairs** in the navigation bar on the left and click on **Create key pair**.

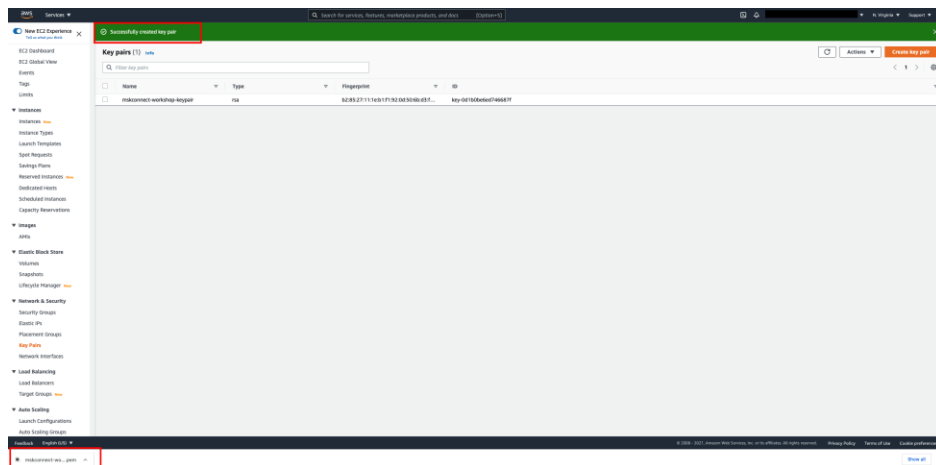


3. Provide a name for the key pair (e.g. **mskconnect-workshop-keypair**) and click on **Create key pair**.



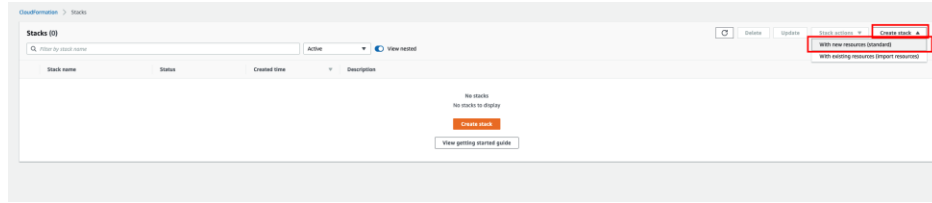
The screenshot shows the AWS Management Console 'Create key pair' page. The 'Name' field is filled with 'mskconnect-workshop-keypair'. The 'Key pair type' is set to 'RSA'. The 'Private key file format' is set to '.pem'. The 'Tags (Optional)' section is empty. The 'Create key pair' button is highlighted with a red box.

4. You should see the message **Successfully created key pair**, and confirm the download of the generated **.pem** file to your local machine.

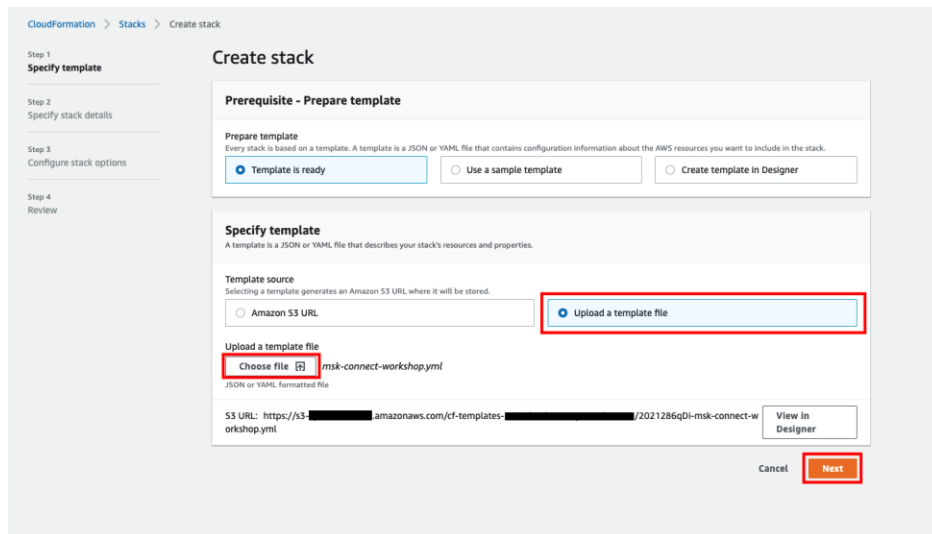


Deploy required AWS resources via CloudFormation

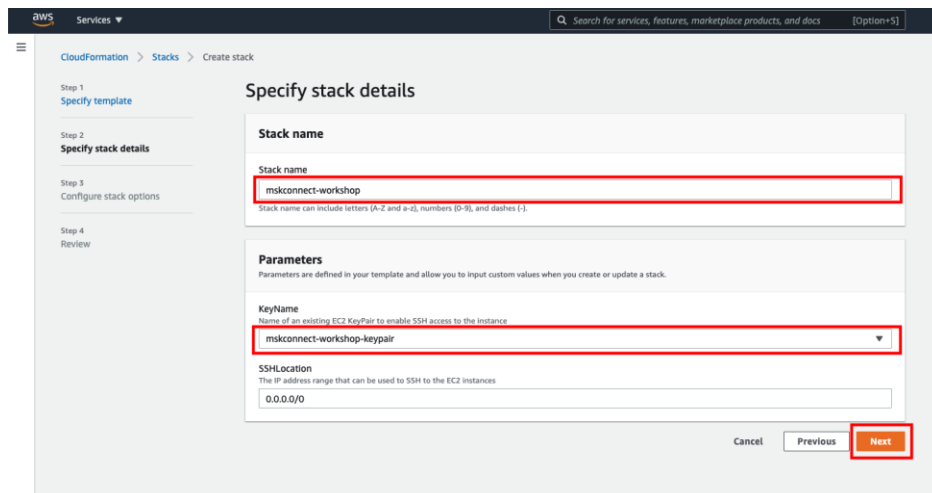
5. Download the CloudFormation template here: [msk-connect-workshop-3.6.0.yml](#)
6. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation/>
7. Click on **Create stack** and select **With new resources (standard)**.



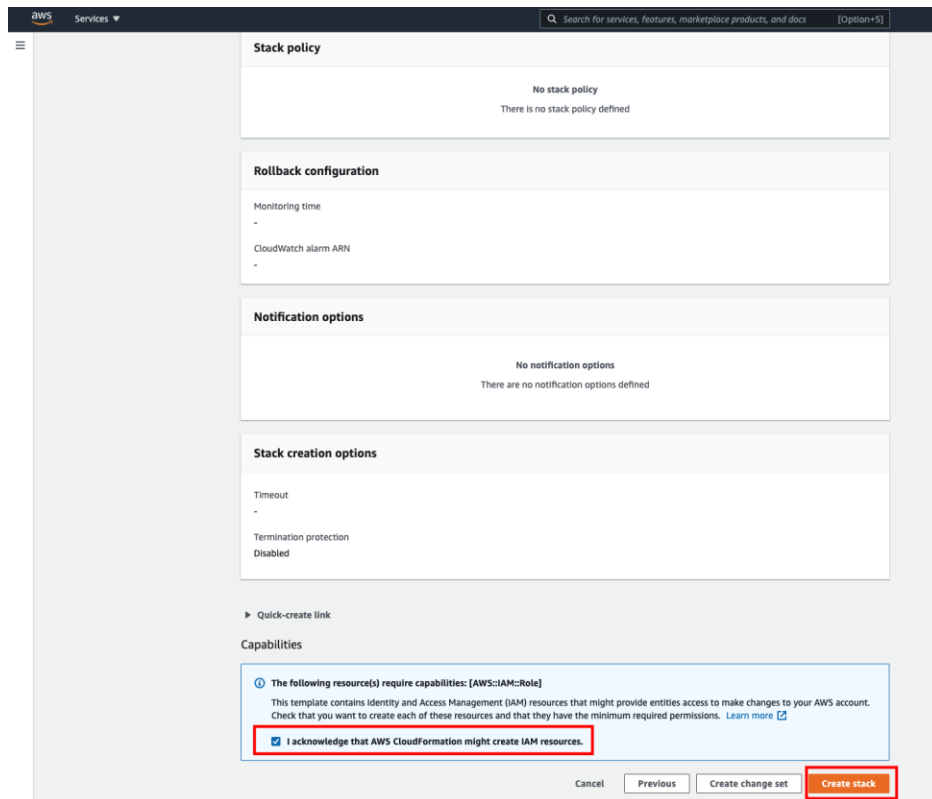
8. Choose **Upload a template file** option and upload the CloudFormation template from your local machine, then click **Next**.



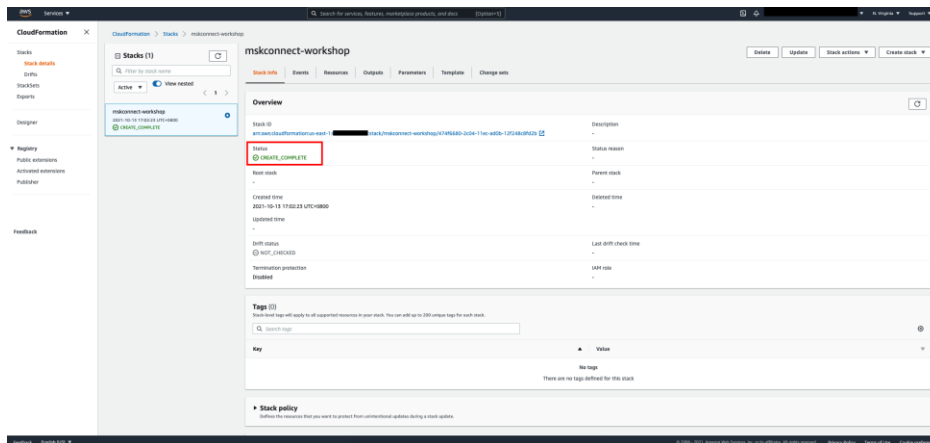
9. Enter **mskconnect-workshop** as the stack name, choose the **mskconnect-workshop-keypair** under **KeyName**, and then click **Next**.



10. Leave the default settings on the step **Configure stack options** and click **Next**.
11. Scroll down to the bottom of the **Review mskconnect-workshop** screen and check the box to acknowledge the creation of IAM resources. Click **Create stack** to trigger the creation of the CloudFormation stack.



12. Wait until the stack creation is completed before proceeding to the next steps. It may take a while for the MSK cluster creation (~30 to 45mins).



What does the above CloudFormation template do?

The CloudFormation stack creates the following resources:

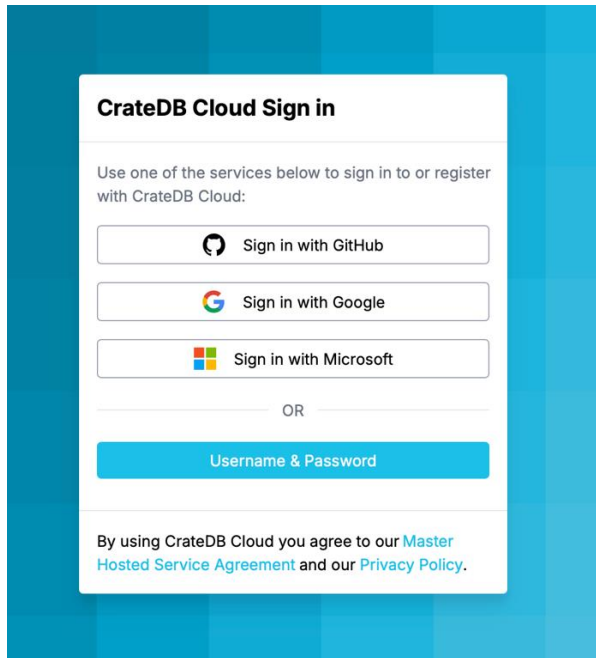
- 2 CloudWatch Log Groups for MSK Cluster and MSK Connect connectors respectively, each with 7 days log retention period
- 1 S3 bucket for storing required Apache Kafka Connect plugin files, and other testing results in some workshop modules
- 1 VPC with 1 Public subnet and 3 Private subnets, and the required networking components such as Route Tables, Internet Gateway, NAT Gateway & Elastic IP, VPC Gateway Endpoint for S3
- 1 m5.large EC2 instance with Java 1.8.0 and Apache Kafka version 3.6.0 installed, and the required Security Group, IAM Role, EC2 Instance Profile
Note: we will not be using this and instead creating a new instance.
- 1 MSK Cluster with 3 kafka.m5.large broker nodes configured with Apache Kafka version 3.6.0, and the required Security Group

Note that the encryption options are disabled in order to reduce the chance of errors in the workshop, this is not a best practice, and you should not use the provided CloudFormation template in any production environments.

Note: While this is running, this would be a good time to create the CrateDB cluster in the next section.

Creating a CrateDB Cluster

1. Go to <https://cratedb.com>
2. Click on the **Start Free** button
3. You have several options to create an account, use the one that is appropriate for you. You won't need to enter any payment details.



4. Once you have an account, log in and deploy a **CRFREE** cluster

Configure your new cluster

Shared
Single-node, suitable for non-production use cases with up to 8 shared vCPUs

Dedicated
Scalable up to 9 nodes, ideal for production workloads with up to 144 vCPUs

Custom
Customizable for large-scale needs, offers any cluster size and custom compute options

Shared tier clusters operate on a single node without replication, sharing vCPUs with other clusters. Performance may vary depending on the overall load, and usage is based on a fair-use principle.
[Learn more about cluster types](#)

Cloud Provider

aws

A

Region

aws AWS US-East (N.Virginia)
eks1.us-east-1.aws

[Request a new region](#)

Node compute size

| | | | | |
|--------|--------------|------------|------------------|--|
| CRFREE | Up to 2 vCPU | 2 GiB RAM | FREE | |
| S2 | Up to 2 vCPU | 2 GiB RAM | \$0.079 per hour | |
| S4 | Up to 3 vCPU | 4 GiB RAM | \$0.158 per hour | |
| S6 | Up to 4 vCPU | 6 GiB RAM | \$0.238 per hour | |
| S12 | Up to 8 vCPU | 12 GiB RAM | \$0.475 per hour | |

Node storage size

8 GiB FREE

- Select a **Shared** cluster (this is the only way to get a free cluster)
- Select **AWS** as the cloud provider

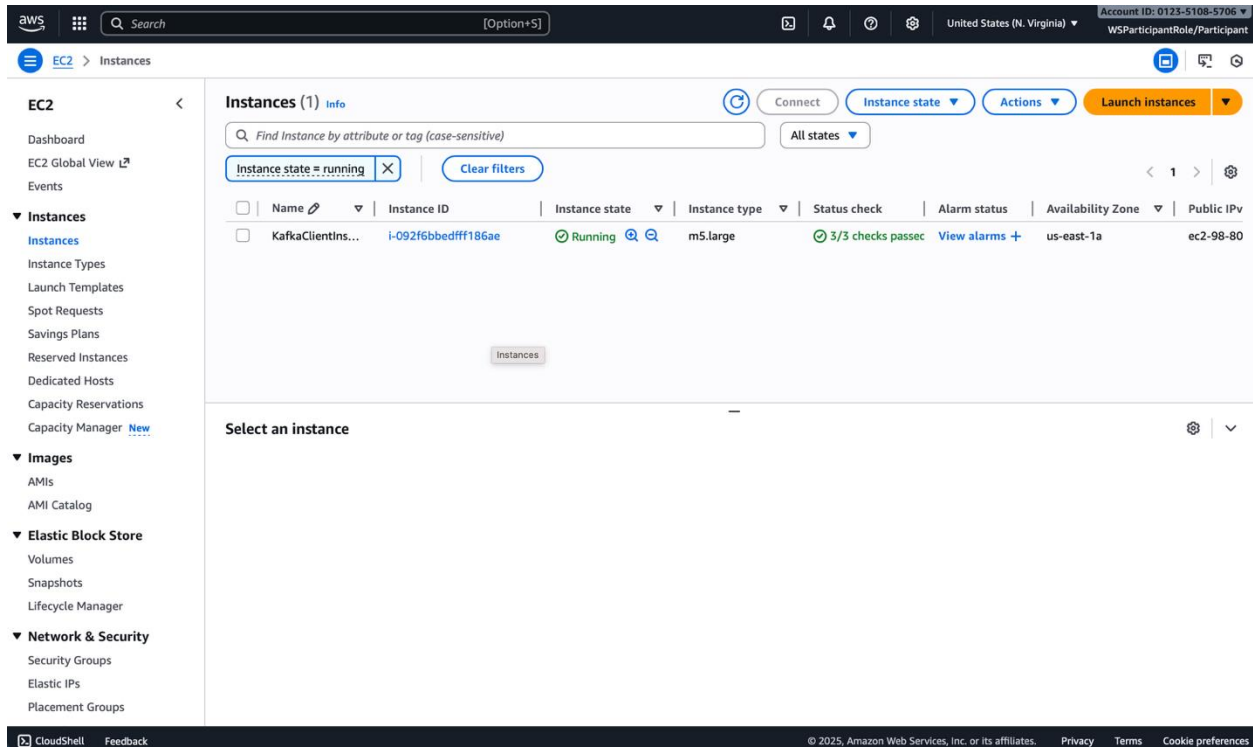
- Select **AWS US-East** as the region (the same as the workshop runs within)
 - Ensure **CRFREE** is selected as compute size
 - Click **Deploy Cluster** to create it, which will take a short amount of time
5. It's **very** important that you save the login credentials, so download or make a note now before moving on.
6. We need to create the destination table for the climate data, run the following in the Console:

```
CREATE TABLE IF NOT EXISTS "demo"."climate_data" (  
  "timestamp" TIMESTAMP WITHOUT TIME ZONE,  
  "geo_location" GEO_POINT,  
  "data" OBJECT(DYNAMIC) AS (  
    "temperature" DOUBLE PRECISION,  
    "u10" DOUBLE PRECISION,  
    "v10" DOUBLE PRECISION,  
    "pressure" DOUBLE PRECISION,  
    "latitude" DOUBLE PRECISION,  
    "longitude" DOUBLE PRECISION,  
    "foo" BIGINT  
  )  
)
```

7. The host information you will need later, so make a note or leave the webpage open for now.

Creating AWS Workshop Environment

Open the EC2 dashboard <https://console.aws.amazon.com/ec2/> go to **Instances**



Note: We need to create a new EC2 instance, do not use the one created as part of the CloudFormation process as it's ARM based, and you will need an x86 instance.

Click the **Launch instances** button and choose the following:

- Name: **cratedb-workshop**
- AMI: **Amazon Linux**
- Architecture: **x86**
- Instance type: **m5a.xlarge**
- Key-pair name: **Choose the one previously created**
- Network Settings: Click on **Edit**
 - VPC: Find and select the workshop **MSKVPC** (same as existing EC2)

▼ Network settings

Info

VPC - required

Info

vpc-0b2c725c46ced30a3 (MSKVPC)

10.0.0.0/16

▲

Q |

vpc-0b5b5297304467acb

172.31.0.0/16

(default)

vpc-0b2c725c46ced30a3 (MSKVPC)

10.0.0.0/16

✓

Auto-assign public IP

Info

Enable

▼

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

Security group name - required

○ Security group: Existing one starting **mskconnect-workshop-KafkaClientInstanceSecurityGroup** (same as existing EC2)

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

Common security groups

Info

Select security groups

▲

Q |

☐ default

sg-00a6722507cce9fe6

VPC: vpc-0b2c725c46ced30a3

☐ mskconnect-workshop-MSKSecurityGroup-MeLw1WbWW9XP

sg-0690e55f9701a8152

VPC: vpc-0b2c725c46ced30a3

☒ mskconnect-workshop-KafkaClientInstanceSecurityGroup-H95RHvuVtzPG

sg-06f6e618d220e70d8

VPC: vpc-0b2c725c46ced30a3

Compare security group rules

work interfaces.

Advanced

- Configure storage: **30GB**
- Ensure it's created in a **public** subnet, not private
- Ensure a public elastic IP is assigned

Create the instance and wait for it to be fully ready before attempting to connect

- Go to the EC2 instance and select the create one
- Click on the **Connect** button and then **Connect** again

EC2 > Instances

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|-------------------|---------------------|----------------|---------------|-------------------|---------------|
| cratedb-works... | i-081e4859677c8424a | Running | m5a.xlarge | Initializing | View alarms + |
| KafkaClientIns... | i-0c4f0fda398d12d35 | Running | m5.large | 3/3 checks passed | View alarms + |

EC2 > Instances > i-081e4859677c8424a > Connect to instance

Connect Info

Connect to an instance using the browser-based client.

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-081e4859677c8424a (cratedb-workshop)

Connection type

☒ Connect using a Public IP
Connect using a public IPv4 or IPv6 address

☐ Connect using a Private IP
Connect using a private IP address and a VPC endpoint

☒ Public IPv4 address
18.212.224.61

☐ IPv6 address
-

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

Installing the demo source code

1. Ensure the git command-line tool is installed:

```
sudo dnf install git
```

2. Clone the repository in the EC2 console (using HTTPS, not SSH). We use a particular “unauthenticated” branch (this removes the SASL authentication code, which is not needed).

```
git clone https://github.com/crate/realtime-demo.git -b unauthenticated
```

3. Change into the directory:

```
cd realtime-demo
```

Running the data producer

To run the producer, we need to set up a virtual Python environment and install dependencies:

```
cd data
python3 -m venv .venv
source .venv/bin/activate
pip3 install -U -r requirements.txt
```

Next, copy the example .env file

```
cp .env.example .env
```

Afterwards change the bootstrap server. Go to **MSK**, click on the MSK Cluster, then click on “View client information”. Then copy one of the Bootstrap servers. Note that they are comma separated. Only select one and make sure it’s fully copied.

aws

Search

[Option+S]

United States (N. Virginia)

Account ID: 7090-4072-9072

WSParticipantRole/Participant

Amazon MSK

Clusters

MSKCluster-22e523d0

Introducing topic management for Amazon MSK clusters

Amazon MSK's new topic management feature lets you view topics directly through the AWS Management Console, streamlining your workflow. [Learn more](#)

0

0

0

0

8

0

MSKCluster-22e523d0

Info

Actions

Cluster summary

Status

Active

User action status

-

Cluster type

Provisioned

Apache Kafka version

3.6.0

ARN

arn:aws:kafka:us-east-1:709040729072:cluster/MSKCluster-22e523d0/f2ebfdb-899d-4db1-9d9b-f304660ca355-6

Last modified

November 24, 2025 at 12:51 UTC

Creation time

November 24, 2025 at 12:51 UTC

View client information

Metrics

Properties

Topics - new

Cluster operations

Cluster alerts (0)

Tags (0)

Lambda

Amazon CloudWatch metrics

Info

Create CloudWatch alarm

Investigate with AI - new

3h 1d 1w 3h

UTC timezone

Explore related

Disk usage by broker

Percent

2e-3

2e-3

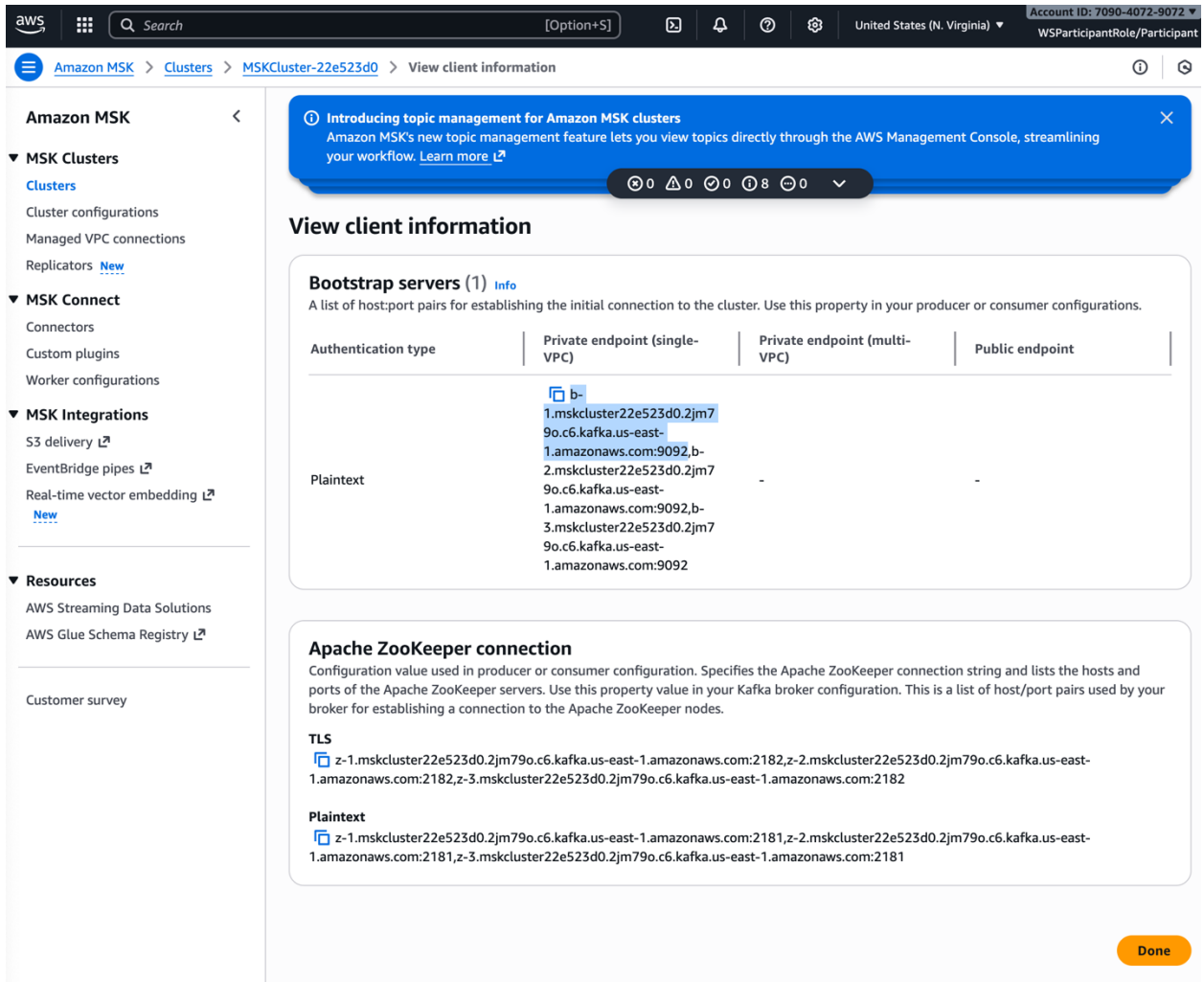
1e-3

CPU (System) usage by broker

Percent

2.5

2



Amazon MSK

- MSK Clusters
 - Clusters
 - Cluster configurations
 - Managed VPC connections
 - Replicators [New](#)
- MSK Connect
 - Connectors
 - Custom plugins
 - Worker configurations
- MSK Integrations
 - S3 delivery [↗](#)
 - EventBridge pipes [↗](#)
 - Real-time vector embedding [↗](#)
 - [New](#)
- Resources
 - AWS Streaming Data Solutions
 - AWS Glue Schema Registry [↗](#)

View client information

Bootstrap servers (1) [Info](#)

A list of host:port pairs for establishing the initial connection to the cluster. Use this property in your producer or consumer configurations.

| Authentication type | Private endpoint (single-VPC) | Private endpoint (multi-VPC) | Public endpoint |
|---------------------|---|------------------------------|-----------------|
| Plaintext | b- 1.mskcluster22e523d0.2jm7 9o.c6.kafka.us-east- 1.amazonaws.com:9092,b- 2.mskcluster22e523d0.2jm7 9o.c6.kafka.us-east- 1.amazonaws.com:9092,b- 3.mskcluster22e523d0.2jm7 9o.c6.kafka.us-east- 1.amazonaws.com:9092 | - | - |

Apache ZooKeeper connection

Configuration value used in producer or consumer configuration. Specifies the Apache ZooKeeper connection string and lists the hosts and ports of the Apache ZooKeeper servers. Use this property value in your Kafka broker configuration. This is a list of host/port pairs used by your broker for establishing a connection to the Apache ZooKeeper nodes.

TLS

z-1.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2182,z-2.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2182,z-3.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2182

Plaintext

z-1.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2181,z-2.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2181,z-3.mskcluster22e523d0.2jm79o.c6.kafka.us-east-1.amazonaws.com:2181

[Done](#)

Next modify the env file in the console with the bootstrap server you copied

vi .env

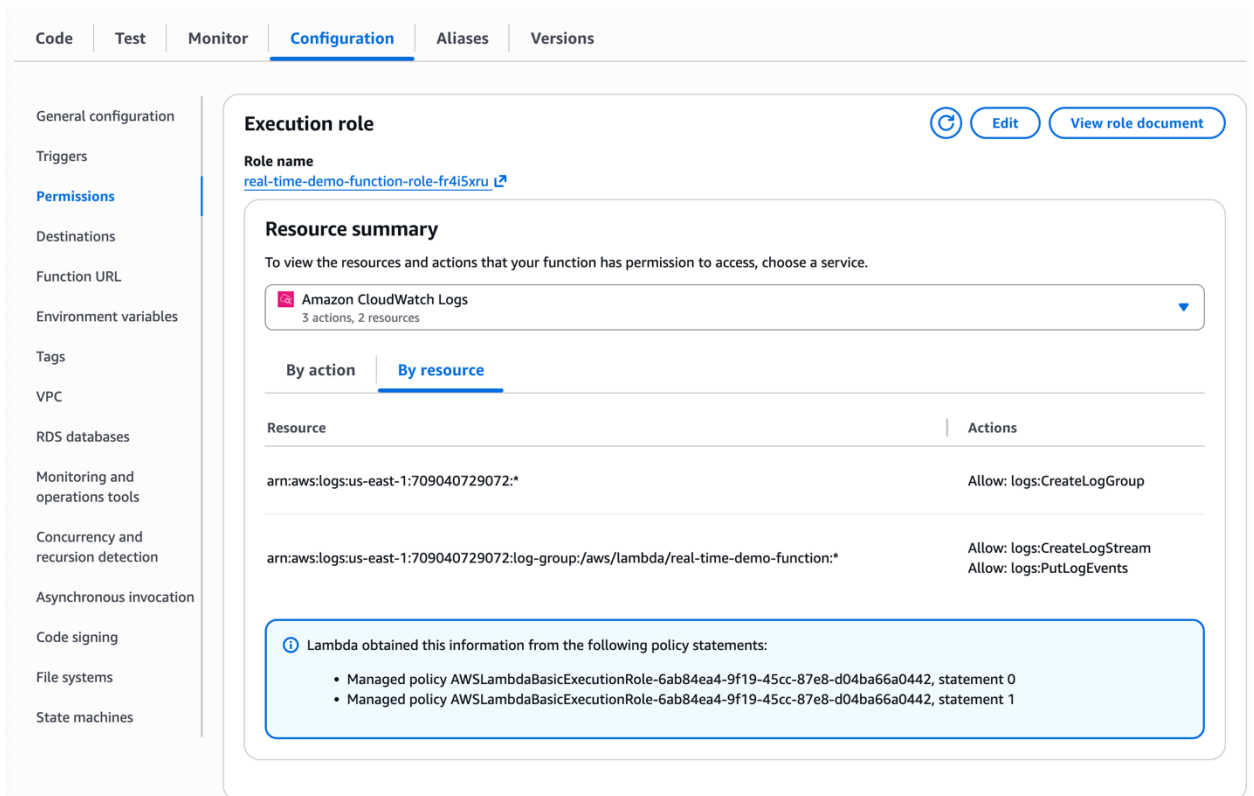
To run the producer, simply execute it.

python3 producer.py

AWS Lambda function for Data Consumer

- Create a Lambda function using the **Author from scratch** option
- Give it the name **real-time-demo-function**
- Choose the Python 3.13 runtime (**not** 3.14)
- Either architecture is fine
- Under **Additional Configurations**:

- Choose the VPC created earlier (MSKVPC)
 - Select the **private** subnets (not public)
 - Select the previously created security group starting with **mskconnect-workshop-MSKSecurityGroup**
- Create the Lambda!
- We need to add some variables
- Go to **Configuration**
- Under **Environment Variables** enter following:
 - CRATEDB_HOST (URL excluding prefix and suffixes, just the hostname)
 - CRATEDB_DB (**crate**)
 - CRATEDB_PASS (password you got when your cratedb cluster was created)
 - CRATEDB_USER (**admin**)
 - CRATEDB_PORT (**4200**)
 - SOURCE_TOPIC (**dev-1-0**) (Note the -0 suffix)
- Some additional security permissions are needed to allow our Lambda to access other resources like the MSK cluster.
 - Go to: **Configuration**, then **Permissions**, then click on the **Role name** at the top to open the Lambda execution role.



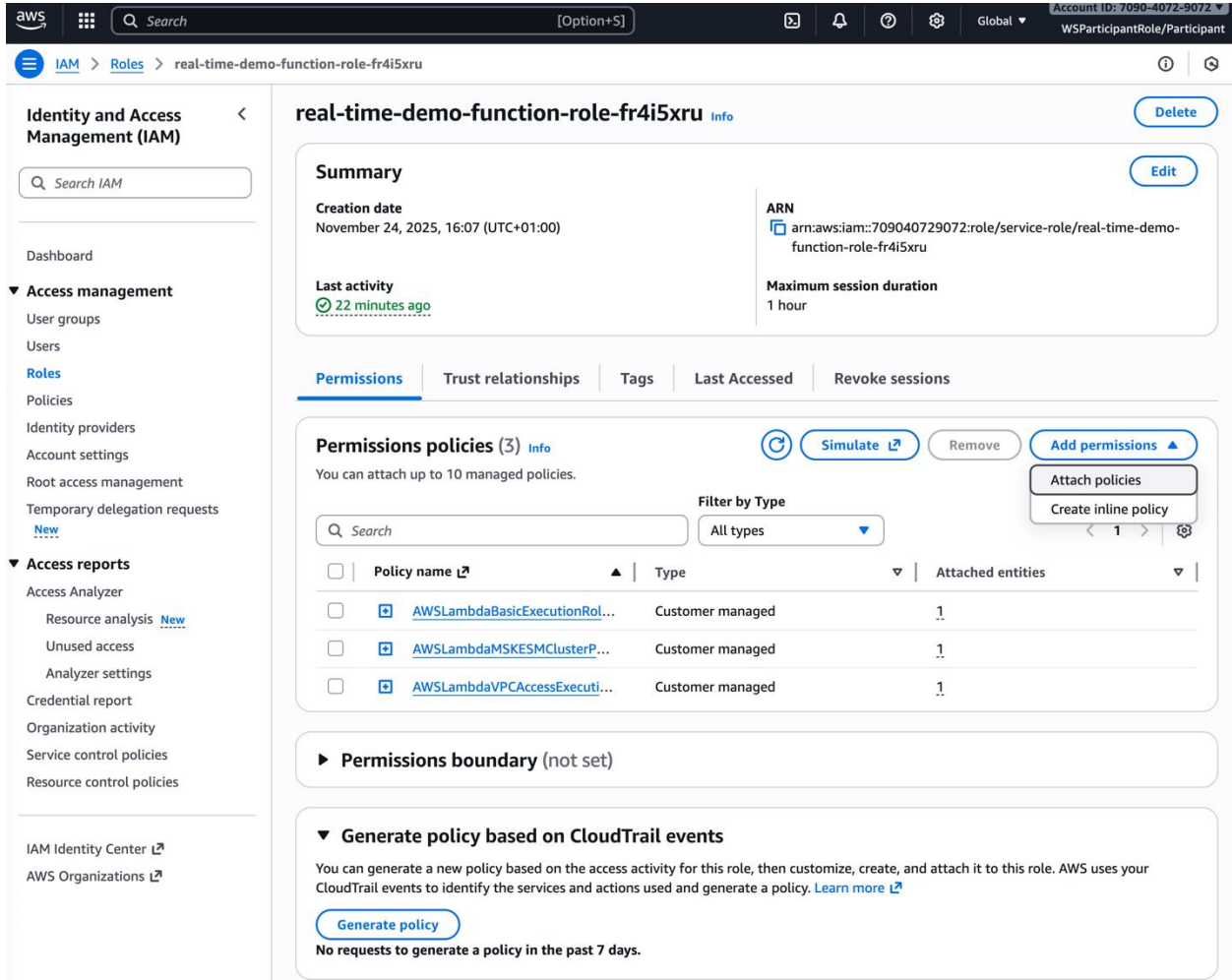
The screenshot shows the AWS Lambda console interface. The 'Configuration' tab is selected. On the left, a sidebar lists various configuration options, with 'Permissions' highlighted. The main panel displays the 'Execution role' configuration for the function 'real-time-demo-function'. The role name is 'real-time-demo-function-role-fr4i5xru'. Under 'Resource summary', 'Amazon CloudWatch Logs' is selected. Below this, a table lists the permissions granted to the role:

| Resource | Actions |
|---|---|
| arn:aws:logs:us-east-1:709040729072:* | Allow: logs:CreateLogGroup |
| arn:aws:logs:us-east-1:709040729072:log-group:/aws/lambda/real-time-demo-function:* | Allow: logs:CreateLogStream Allow: logs:PutLogEvents |

At the bottom, a note states: 'Lambda obtained this information from the following policy statements:'

- Managed policy AWSLambdaBasicExecutionRole-6ab84ea4-9f19-45cc-87e8-d04ba66a0442, statement 0
- Managed policy AWSLambdaBasicExecutionRole-6ab84ea4-9f19-45cc-87e8-d04ba66a0442, statement 1

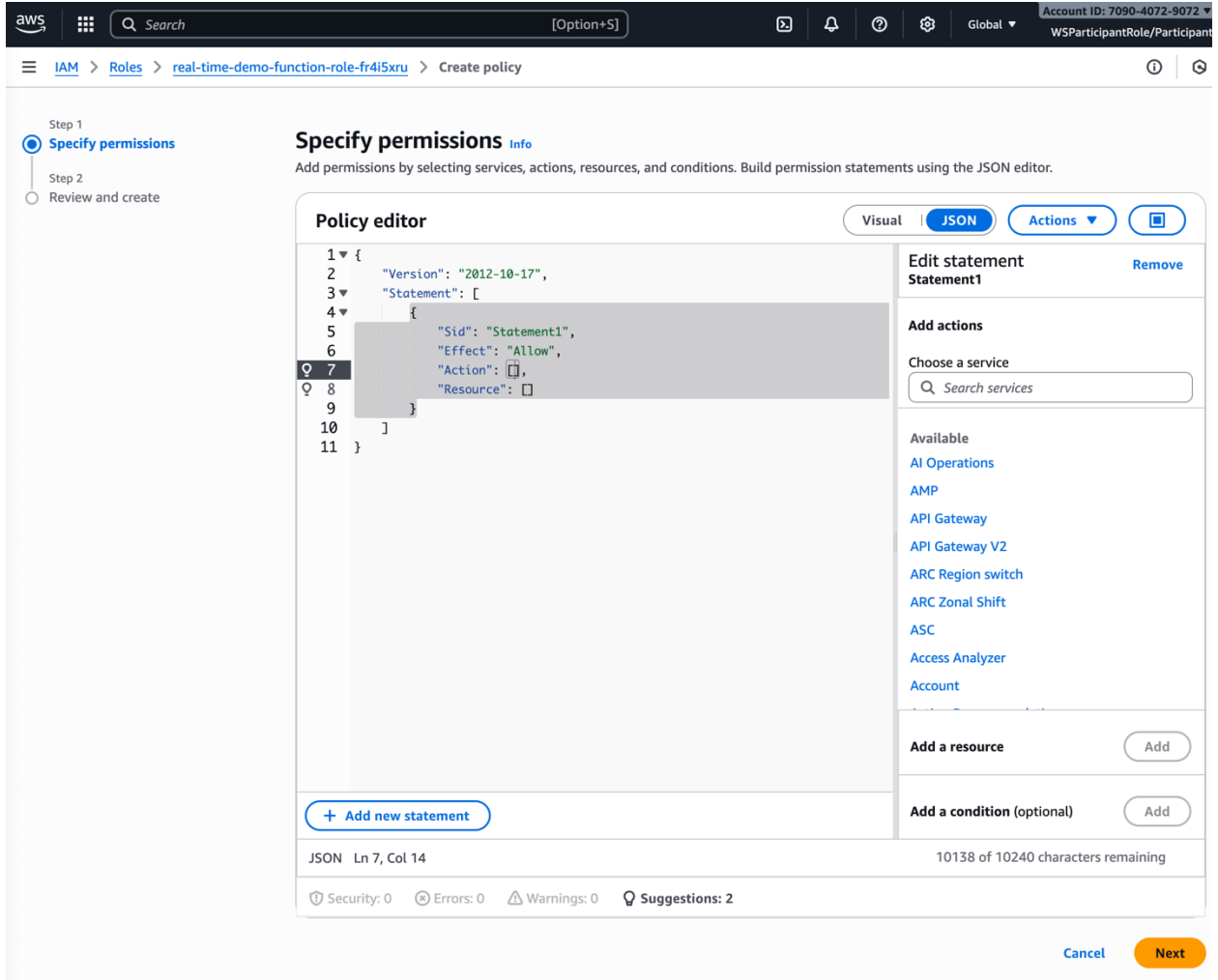
- Add the following permissions to the Lambda execution role:
 - **AdministratorAccess** (never do this in a production environment!)



The screenshot shows the AWS IAM console interface for the role `real-time-demo-function-role-fr4i5xru`. The left sidebar contains navigation links for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays the role's summary, including its creation date (November 24, 2025, 16:07 UTC+01:00), ARN, and last activity (22 minutes ago). The 'Permissions' tab is active, showing a list of three attached policies. The 'Add permissions' button is highlighted, and a dropdown menu is open, showing options to 'Attach policies' or 'Create inline policy'.

| Policy name | Type | Attached entities |
|---|------------------|-------------------|
| AWSLambdaBasicExecutionRol... | Customer managed | 1 |
| AWSLambdaMSKESMClusterP... | Customer managed | 1 |
| AWSLambdaVPCLAccessExecuti... | Customer managed | 1 |

- After that add a new inline policy to allow the Lambda execution role to access MSK (this rule allows access to all MSK clusters)



Step 1
 Specify permissions
 Step 2
 Review and create

Specify permissions [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual
JSON
Actions

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Action": [
8         "kafka:DescribeCluster",
9         "kafka:DescribeClusterV2",
10        "kafka:GetBootstrapBrokers"
11      ],
12       "Resource": "arn:aws:kafka:*:*:cluster/*"
13     }
14   ]
15 }

```

+ Add new statement

Edit statement
Statement1 [Remove](#)

Add actions

Choose a service

Available
AI Operations
AMP
API Gateway
API Gateway V2
ARC Region switch
ARC Zonal Shift
ASC
Access Analyzer
Account

Add a resource [Add](#)

Add a condition (optional) [Add](#)

JSON Ln 7, Col 14
10138 of 10240 characters remaining

Security: 0 Errors: 0 Warnings: 0 Suggestions: 2

[Cancel](#)
[Next](#)

- Select JSON and paste the code below

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka:GetBootstrapBrokers"
      ],
      "Resource": "arn:aws:kafka:*:*:cluster/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:Connect",
        "kafka-cluster:AlterCluster",
        "kafka-cluster:DescribeCluster",
        "kafka-cluster:DescribeTopic",
        "kafka-cluster:ReadData",
        "kafka-cluster:AlterGroup",
        "kafka-cluster:DescribeGroup"
      ]
    }
  ]
}

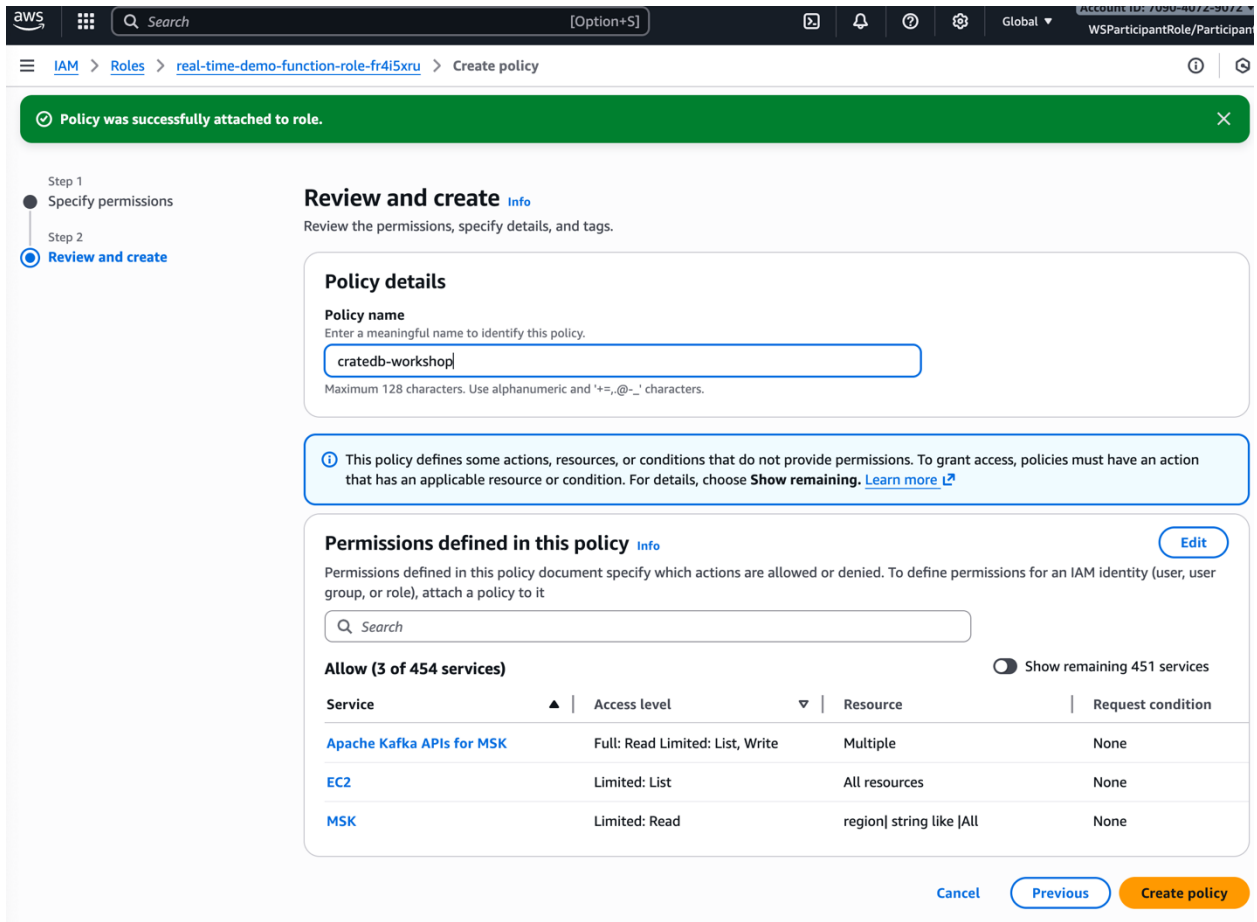
```

```

    ],
    "Resource": [
      "arn:aws:kafka:*:*:cluster/*",
      "arn:aws:kafka:*:*:topic/*",
      "arn:aws:kafka:*:*:group/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  }
]
}

```

- Click on Next and name this **cratedb-workshop**
- Create the policy



The screenshot shows the AWS IAM console interface during the 'Review and create' step of creating a new policy. A green banner at the top indicates 'Policy was successfully attached to role.' The left sidebar shows two steps: 'Specify permissions' and 'Review and create', with the latter being the active step. The main content area is titled 'Review and create' and includes a sub-header 'Policy details'. A text input field for 'Policy name' contains 'cratedb-workshop'. Below this, a blue information box states: 'This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining**. [Learn more](#)'. The 'Permissions defined in this policy' section shows a search bar and a table of permissions. The table has columns for Service, Access level, Resource, and Request condition. It lists three services: 'Apache Kafka APIs for MSK', 'EC2', and 'MSK'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create policy'.

Policy details

Policy name
Enter a meaningful name to identify this policy.

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

Permissions defined in this policy

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining**. [Learn more](#)

Permissions defined in this policy

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (3 of 454 services) ☒ Show remaining 451 services

| Service | Access level | Resource | Request condition |
|---------------------------|---------------------------------|--------------------------|-------------------|
| Apache Kafka APIs for MSK | Full: Read Limited: List, Write | Multiple | None |
| EC2 | Limited: List | All resources | None |
| MSK | Limited: Read | region string like [All | None |

Cancel Previous **Create policy**

Now we upload the Lambda code, this has been provided in the form of a ZIP file release to make it simple to update.

- The link is: <https://github.com/crate/realtime-demo/releases/tag/Resources>
- Once downloaded go to the **Lambda** page and select the **Code** section
- Go to **Upload from** button which allows the ZIP file to be uploaded.

Create a trigger for the AWS Lambda function

Next step is to create a trigger from MSK when a new value in our demo topic is created. To do this:

- Go to created Lambda Function
- Go to **Configuration**
- Go to **Triggers**
- Select **Add Trigger**
- Select **MSK** as the source
- Find the cluster created earlier (likely named real-time-demo)
- The topic name is **dev-1**
- **Provisioned mode** should be **disabled**
- Use **IAM** authentication
- Set **Starting position** to **At timestamp**
- Starting position is timestamp **2025-10-14**
- Under additional settings, batch size determines how quickly data is ingested, set to **100**
- Click on **Add**

Installing Grafana onto AWS EC2 instance

Update packages

```
sudo dnf update -y
```

Add the Grafana repository

```
sudo tee /etc/yum.repos.d/grafana.repo <<EOF
[grafana]
name=Grafana OSS
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
EOF
```

Install Grafana

```
sudo dnf install grafana -y
```

Enable and start the service

```
sudo systemctl enable grafana-server  
sudo systemctl start grafana-server
```

Open Grafana port

Edit the security group for the EC2 instance and open port 3000 for the Grafana dashboards.

- Go to EC2
- Select the running workshop instance

Instance summary for i-081e4859677c8424a (cratedb-workshop) [Info](#)

[Refresh](#)
[Connect](#)
[Instance state ▼](#)
[Actions ▼](#)

Updated 1 minute ago

| | | |
|--|--|--|
| Instance ID i-081e4859677c8424a | Public IPv4 address 18.212.224.61 open address | Private IPv4 addresses 10.0.0.32 |
| IPv6 address - | Instance state Running | Public DNS ec2-18-212-224-61.compute-1.amazonaws.com open address |
| Hostname type IP name: ip-10-0-0-32.ec2.internal | Private IP DNS name (IPv4 only) ip-10-0-0-32.ec2.internal | Elastic IP addresses - |
| Answer private resource DNS name - | Instance type m5a.xlarge | AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more |
| Auto-assigned IP address 18.212.224.61 [Public IP] | VPC ID vpc-0b2c725c46ced30a3 (MSKVPC) | Auto Scaling Group name - |
| IAM Role - | Subnet ID subnet-08cf7c34cf515d1d3 (MSKPublicSubnet) | Managed false |
| IMDSv2 Required | Instance ARN arn:aws:ec2:us-east-1:709040729072:instance/i-081e4859677c8424a | |
| Operator - | | |

[Details](#)
[Status and alarms](#)
[Monitoring](#)
[Security](#)
[Networking](#)
[Storage](#)
[Tags](#)

▼ Security details

| | | |
|--|---------------------------------|--|
| IAM Role - | Owner ID 709040729072 | Launch time Mon Nov 24 2025 14:35:27 GMT+0100 (Central European Standard Time) |
| Security groups sg-06fbe618d220e70d8 (mskconnect-workshop-KafkaClientInstanceSecurityGroup-H95RHvuVtzPG) | | |

- Click on the security groups
- Click on **Edit inbound rules**
- Click on **Add rule**
- Select:
 - Type: Customer TCP
 - Port range: 3000
 - Source: Anywhere-IPv4

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Security group rule ID | Type | Protocol | Port range | Source | Description - optional | |
|------------------------|------------|----------|------------|--------|------------------------|------------------------|
| sgr-030cad363bd5cd20e | SSH | TCP | 22 | Cus... | | Delete |
| - | Custom TCP | TCP | 3000 | An... | | Delete |

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

- Click on **Save rules**
- Go back to EC2
- Select the running workshop instance
- Make note of the public IPv4 address of the EC2 instance

Import dashboards

Your dashboard should be accessible at <http://ip:3000> (where IP is the assigned external IP address of the EC2 instance).

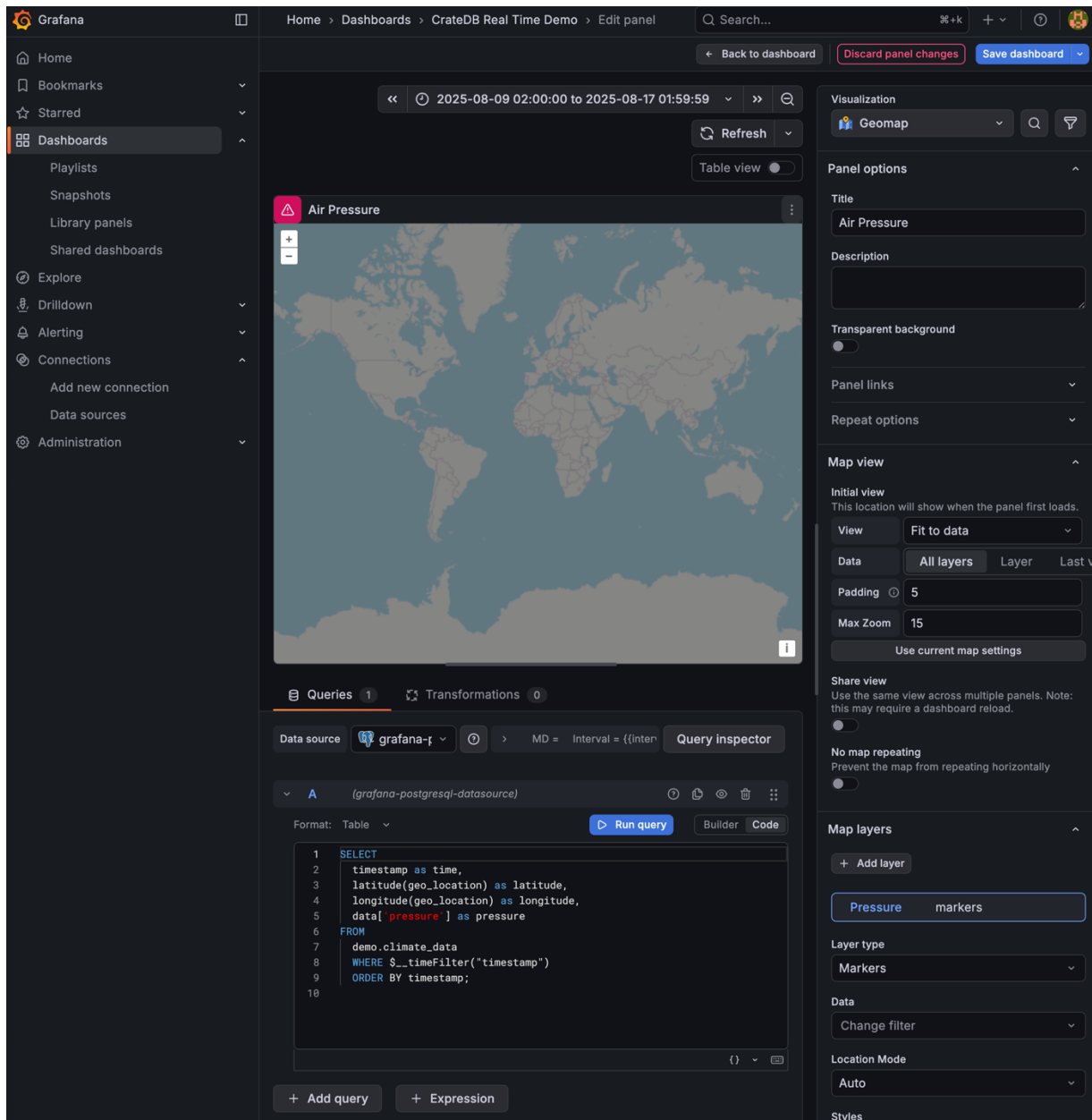
To login, the default Grafana username and passwords are admin/admin, then either skip or set a new password.

Download both dashboards from <https://github.com/crate/realtime-demo/releases/tag/Dashboards> and use the import functionality in Grafana.

- Go to **Dashboards**
- Click on **New** and select **Import**
- Select one of the downloaded two JSON documents
- Click on **Import**
- Upload the second one as well

Initially, they will be in an error state, that's because we need to configure the data source.

- Go to **Connections** on the menu on the left
- Click on **Add new connection**
- Search for **PostgreSQL** and add those details from the beginning
 - Name: **grafana-postgresql-datasource**
 - Host URL
 - Database name: **crate**
 - Username: **admin**
 - Password: (password from the CrateDB instance)
- Save and test
- Now go into the dashboards and edit each of the controls on each page, changing the datasource to the newly created one. A **refresh** and the graphs should populate.



The screenshot shows the Grafana interface for editing a dashboard panel titled "Air Pressure". The panel displays a world map visualization. The left sidebar contains navigation options like Home, Bookmarks, Starred, Dashboards, Playlists, Snapshots, Library panels, Shared dashboards, Explore, Drilldown, Alerting, Connections, Add new connection, Data sources, and Administration.

The top navigation bar shows the path: Home > Dashboards > CrateDB Real Time Demo > Edit panel. It includes a search bar, a "Back to dashboard" button, and buttons for "Discard panel changes" and "Save dashboard".

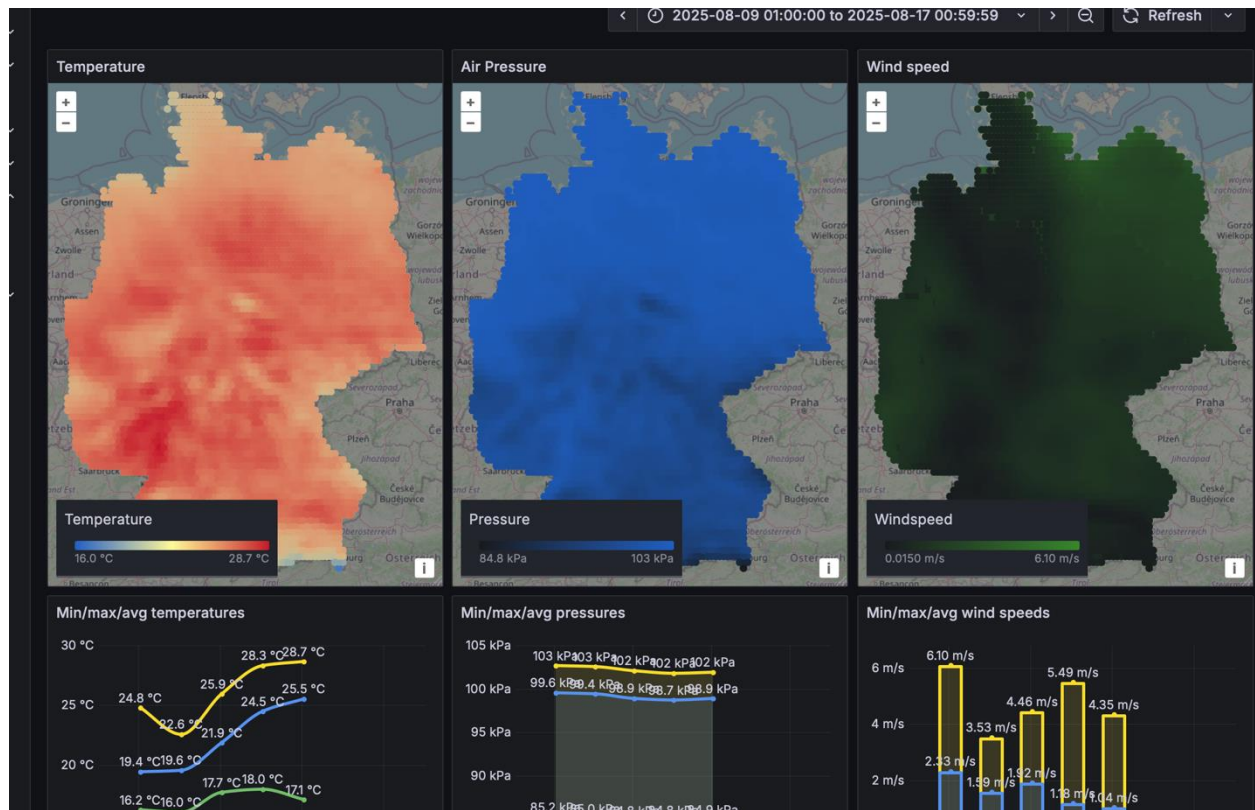
The main panel area shows a world map visualization. Above the map, there is a time range selector set to "2025-08-09 02:00:00 to 2025-08-17 01:59:59", a "Refresh" button, and a "Table view" toggle.

The right sidebar contains the "Panel options" section, which includes fields for "Title" (Air Pressure), "Description", "Transparent background" (toggle), "Panel links", "Repeat options", "Map view" (Initial view, View: Fit to data, Data: All layers, Layer, Last v), "Padding" (5), "Max Zoom" (15), "Use current map settings", "Share view", "No map repeating" (toggle), and "Map layers" (Add layer, Pressure markers, Layer type: Markers, Data: Change filter, Location Mode: Auto, Styles).

The bottom section shows the "Queries" tab with a single query for the "grafana-postgresql-datasource". The query is as follows:

```
1 SELECT
2   timestamp as time,
3   latitude(geo_location) as latitude,
4   longitude(geo_location) as longitude,
5   data[ 'pressure' ] as pressure
6 FROM
7   demo.climate_data
8 WHERE $__timeFilter("timestamp")
9 ORDER BY timestamp;
```

Below the query editor are buttons for "+ Add query" and "+ Expression".



All complete!