

# Setting up observability for fleets

Last updated 2025-10-14

Logging and monitoring can help you troubleshoot issues in IBM Cloud® Code Engine and can provide insight into your workload performance. Setting up logging and monitoring for fleets is done at the project level and is different from other Code Engine features.

- [Setting up logging for fleets](#)
- [Setting up monitoring for fleets](#)

1

**Note:** Logging and monitoring are optional, and can be set up any time after a fleet is created.

## Setting up logging for fleets

Logging for fleets is set up at the project level. After you perform these set up steps for a project, logging is available for all fleets in that project.

### 1. Make sure you have an existing Cloud Logs service instance

You must have a Cloud Logs service instance that targets the region of the Code Engine project that your fleets run in. To check your existing Cloud Logs service instances, navigate to the [Logging routing page](#) in the UI and identify the instance that targets the relevant region. If the region does not have a targeted logging instance, follow the UI prompts to set one.

### 2. Set the permissions for the Cloud Logs instance

Make sure you have a service ID and an API key that assign access to the Cloud Logs instance and set the required permissions.

- 1 [Create a service ID](#). Then assign the service ID [an access policy](#) that targets the Cloud Logs instance in your account. Assign this access policy the **Sender** service access level.
- 2 [Create an API key for the service ID](#). Save the API key in a secure location so you can reference it later.

### 3. Get the private Ingress endpoint for the Cloud Logs instance

Get the private Ingress endpoint for the Cloud Logs instance.

- 1 Navigate to the [Logging instances page](#) in the UI and click on the relevant instance.
- 2 Click **Endpoints**.
- 3 Find the private Ingress endpoint. Save this value in a secure location so you can reference it later.

### 4. Create a virtual private endpoint gateway that connects to the logging instance

Create a virtual private endpoint gateway that connects to the logging instance. This VPE gateway must be created in the VPC that your fleets run in.

- 1 Navigate to the [VPE gateways page](#) in the console.
- 2 Click **Create**.
- 3 Select the location and region that your Cloud Logs instance exists in.
- 4 Select the VPC that your fleets run in.
- 5 Connect the service to your Cloud Logs instance.
  - a. In the **Request connection to a service** section, select the option for **IBM Cloud Service**.
  - b. Under **Cloud service offerings**, select **Cloud Logs**.
  - c. Under **Cloud service regions**, select the region that your Cloud Logs instance exists in.
  - d. From the table, select the relevant Cloud Logs instance to connect to the VPE.
- 6 Click **Create**.

### 5. Create reserved IP addresses in the VPE gateway

For each zone in the VPE, add a reserved IP address to one subnet.

- 1 Navigate to the [VPE gateways page](#) in the console.
- 2 Click on the VPE you created in the previous step.
- 3 Click **Manage attached resources**.
- 4 Click **Reserve or bind IP**.
- 5 Follow the prompts to reserve an IP address for a subnet in one of the VPE zones.
- 6 Repeat this process for one subnet in each VPE zone.

### 6. Add the logging values to the fleets default secret

Add the required logging values to the fleets default secret.

- 1 In the console, go to the [Code Engine projects page](#) and select the project that your fleets run in.
- 2 Click **Secrets and configmaps**.
- 3 Click on the `codeengine-fleet-defaults` secret.
- 4 Add the key-value pairs listed in the table below. Note that the `logging_sender_api_key` and `logging_ingress_endpoint` were saved in earlier steps.

| Key                                   | Value  |
|---------------------------------------|--|
| <code>logging_ingress_endpoint</code> | The private Ingress endpoint for the Cloud Logs instance. This value is saved in a <a href="#">previous step</a> .   |
| <code>logging_sender_api_key</code>   | The API key for the service ID that targets the Cloud Logs instance. This value is saved in a <a href="#">previous step</a> .  |
| <code>logging_ingress_port</code>     | The port used to connect to the Cloud Logs instance. For private VPEs, specify <code>443</code> .  |
| <code>logging_level_worker</code>     | The level of details you want to include in your worker logs. Accepted levels include: <code>DEBUG</code> , <code>INFO</code> , <code>WARN</code> , and <code>ERROR</code> . Default level: <code>WARN</code> .      |
| <code>logging_level_agent</code>      | The level of details you want to include in your fleet agent logs. Accepted levels include: <code>DEBUG</code> , <code>INFO</code> , <code>WARN</code> , and <code>ERROR</code> . Default level: <code>WARN</code> . |

Table 1. Logging key-value pairs for the fleets default secret.

### 7. View your fleets logs

Now that logging is set up for fleets in your Code Engine project, you can view your fleet logs. See [Viewing logs and monitoring data for fleets](#) for steps.