

After you [set up debugging](/debugger/docs/setup) in Stackdriver Debugger and deploy or start your app, you can take or view debug snapshots in the [source console](https://source.cloud.google.com).

Snapshots capture local variables and the call stack at a specific line location in your app's source code. You can return a snapshot of your app's state based on conditions and line locations you specify. The snapshot details can help you debug your code.

A few seconds after you set the line location, the snapshot settings are applied to all running instances of your app. The first time any instance runs the code at the snapshot location, Debugger takes a snapshot and makes it available for viewing. Debugger implements several strategies for minimizing latency caused when capturing data.

Debugger only takes a snapshot *once* during app runtime. You can manually [retake the snapshot](#) to capture a new set of data.

Snapshots might not be available during startup while the debugger initializes.

1. In the Google Cloud Console, open Cloud Source Repositories.

[Open Cloud Source Repositories](https://source.cloud.google.com/repos)

The **All repositories** page opens. Alternatively, you can open the [My source](/source-repositories/docs/browsing-repositories#browsing_to_a_repository) view.

2. Click the name of a repository and go to the file that contains source code you want to watch.
3. Click the line number of the source code location.

```

80 def draw_mandelbrot(width, height,
81                     left, right, top, bottom,
82                     iterations,
83                     z0):
84     """Returns a PIL.Image representing the given portion of the set."""
85     x, y = numpy.meshgrid(numpy.linspace(left, right, width),
86                          numpy.linspace(top, bottom, height))
87     c = x + 1j*y
88     z = c.copy()
89     z[:] = z0

```

4. When prompted, select the app you want to debug, and then click **Continue**.

Select an application to start debugging

Select an application ▼

i If you can't see your application, verify that it's associated with a version of this repository. [Learn more](#) ↗

Cancel
Continue

5. Click **Create snapshot**.

[Create snapshot](#)
Capture variables at each point in the call stack

[Create logpoint](#)
Generate a message each time a line is executed

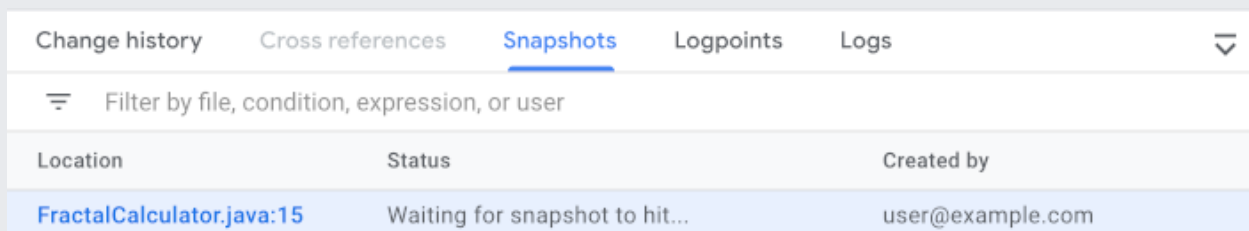
The new snapshot appears in an adjacent pane of the Google Cloud Console.

Change history	Cross references	Snapshots	Logpoints	Logs
☰ Filter by file, condition, expression, or user				
Location	Status	Created by		
FractalCalculator.java:15	Waiting for snapshot to hit...	user@example.com		

A snapshot condition is an expression that must evaluate to `true` for the snapshot to be taken. Expressions in Java, Python, and Go are supported. Snapshot conditions are evaluated each time any instance runs the line until the condition evaluates to `true` or the snapshot times out.

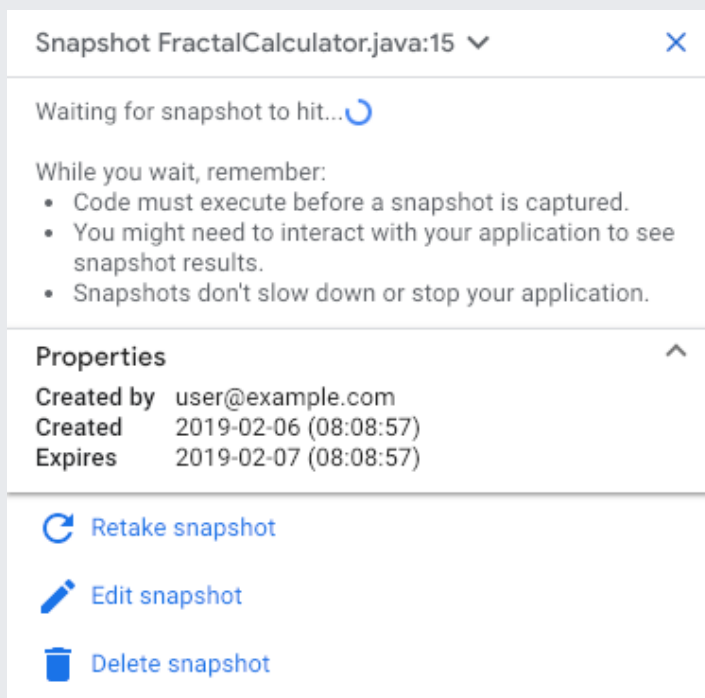
The condition is a full boolean expression that can include logical operators:

1. In the GCP Console, click the **Snapshots** tab, and then select a snapshot.



Change history	Cross references	Snapshots	Logpoints	Logs
Filter by file, condition, expression, or user				
Location	Status	Created by		
FractalCalculator.java:15	Waiting for snapshot to hit...	user@example.com		

2. After the **Snapshot** panel opens, click **Edit snapshot**.



Snapshot FractalCalculator.java:15

Waiting for snapshot to hit...

While you wait, remember:

- Code must execute before a snapshot is captured.
- You might need to interact with your application to see snapshot results.
- Snapshots don't slow down or stop your application.

Properties

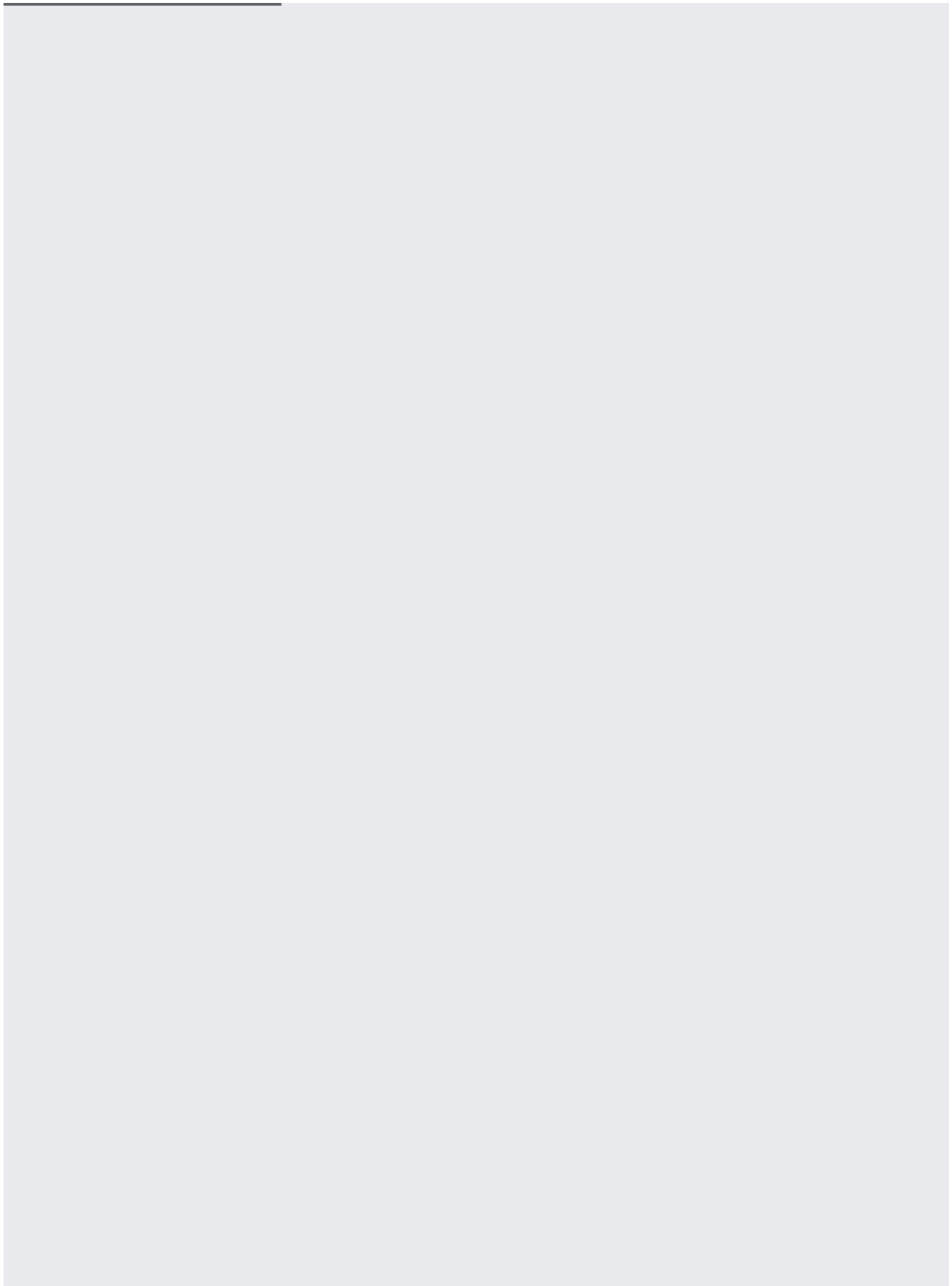
Created by user@example.com
Created 2019-02-06 (08:08:57)
Expires 2019-02-07 (08:08:57)

Retake snapshot
Edit snapshot
Delete snapshot

3. In the **Condition** field, enter the snapshot condition.

4. Click **Update**.

You can use the following language features to express conditions.

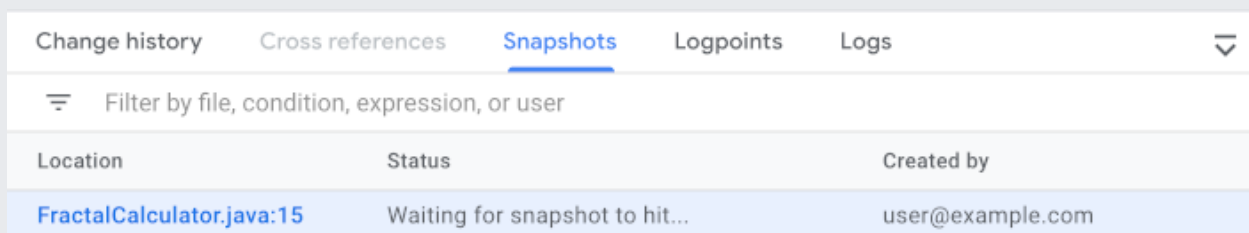


Snapshot expressions let you evaluate complex expressions or traverse object hierarchies when a snapshot is taken. Expressions support the same language features as [snapshot conditions](#) (`#snapshot_conditions`) described above.

The following are typical uses for expressions:

- Viewing static or global variables that are not part of the local variable set.
- Viewing deeply nested member variables without having to expand a local variable in the debugger panel every time.
- Avoiding repetitive mathematical calculations. For example, calculating a duration in seconds with `(endTimeMillis - startTimeMillis) / 1000.0`.

1. In the lower pane of the GCP Console, click the **Snapshots** tab, and then select a snapshot.



Change history	Cross references	Snapshots	Logpoints	Logs
Filter by file, condition, expression, or user				
Location	Status	Created by		
FractalCalculator.java:15	Waiting for snapshot to hit...	user@example.com		

2. In the panel that opens, click **Edit snapshot**.

Snapshot FractalCalculator.java:15
✕

Waiting for snapshot to hit...

While you wait, remember:

- Code must execute before a snapshot is captured.
- You might need to interact with your application to see snapshot results.
- Snapshots don't slow down or stop your application.

Properties ^

Created by user@example.com
 Created 2019-02-06 (08:08:57)
 Expires 2019-02-07 (08:08:57)

Retake snapshot

Edit snapshot

Delete snapshot

3. In the **Expression** field, enter the expression.

If you want to add expressions, press the Tab key.

4. Click **Update**.

Snapshot data appears in the source console after the app runs the code at the location you specified.

1. In the lower pane of the GCP Console, click the **Snapshots** tab, and then select a snapshot.

Location	Status	Created by
FractalCalculator.java:15	Waiting for snapshot to hit...	user@example.com

A panel containing snapshot details displays values of instance variables and the stack trace at the snapshot code location.

Snapshot FractalCalculator.java:15

Properties

Created by user@example.com
Created 2019-02-06 (08:08:57)
Captured 2019-02-06 (08:20:58)

Variables

▶ this
x -0.849842
y -0.77443
iterationsCount 0
curRe 0



Call Stack


com.google.cloud.debugger.mandelbrot.FractalCalcul...
FractalCalculator.java:15
com.google.cloud.debugger.mandelbrot.GeneratorServ...
GeneratorServlet.java:54
javax.servlet.http.HttpServlet.service
HttpServlet.java:618
javax.servlet.http.HttpServlet.service
HttpServlet.java:725
org.eclipse.jetty.servlet.ServletHolder.handle
ServletHolder.java:808
org.eclipse.jetty.servlet.ServletHandler.CachedCha...
ServletHandler.java:1669
com.google.apphosting.utils.servlet.ParseBlobUploa...
ParseBlobUploadFilter.java:125
org.eclipse.jetty.servlet.ServletHandler.CachedCha...

You can examine the value of local variables at the moment the snapshot was taken and drill down into deeper data structures. You can also click on any call stack frame and examine the local variables at that level in the stack.

A snapshot is only taken once. To capture another snapshot of your app state at the same location, follow these steps:

1. In the **Snapshots** panel of the GCP Console, hold the pointer over the snapshot.
2. Click **Retake**.

Change history	Cross references	Snapshots	Logpoints	Logs	⌵
☰ Filter by file, condition, expression, or user					
Location	Status	Created by			
FractalCalculator.java:15	Captured at 2019-02-06 (08:20:58)	user@example.com			

1. In the **Snapshots** panel of the GCP Console, hold the pointer over the snapshot.
2. Click **Delete** .

You can share a snapshot with another project member by sharing the snapshot URL from your browser. You can also save this URL for future reference and come back to it again to view its results. Debugger uses a new URL for each snapshot taken. This lets you share distinct sets of results even if they were captured at the same location in the code.

- [Learn more about Debugger \(/debugger/docs/\)](/debugger/docs/).