Support (https://cloud.google.com/support/) Documentation

Best Practices for Working with Cloud Support

Writing a detailed support case makes it easier for the Google Support team to respond to you quickly and efficiently. When your support case is missing critical details, we need to ask for more information, which takes additional time. This guide lets you know the information we need to resolve your technical support case, so we can resolve it faster.

Describing the issue

The best issue reports are both detailed and specific. They tell us what happened and what you expected to happen. A good issue report contains the following details:

- **Time:** The specific timestamp when the issue began.
- **Product:** The product(s) and feature(s) associated with the issue.
- **Location:** The zone(s) where you are seeing the issue.
- **Identifiers:** The project/application ID and other concrete identifiers that help us research the issue.
- **Useful artifacts:** Any details you can provide to help us diagnose the issue.

The following sections describe these concepts in greater detail.

Time

Using the ISO 8601 (https://en.wikipedia.org/wiki/ISO_8601) format for the date and time stamp, tell us when you first noticed this issue and how long it lasted.

Examples:

- Starting at 2017-09-08T15:13:06+00:00 and ending 5 minutes later, we observed ...
- Observed intermittently, starting no earlier than 2017-09-10 and observed 2-5 times...
- Ongoing since 2017-09-08T15:13:06+00:00...
- From 2017-09-08T15:13:06+00:00 to 2017-09-08T15:18:16+00:00...

The Support Engineer resolving the issue is very likely not in your timezone, so relative statements like the following make the problem harder to diagnose:

- "This started sometime yesterday..." (Forces us to infer the implied date.)
- "We noticed the issue on 9/8..." (Ambiguous, as some may interpret this as September 8, and others may interpret it as August 9.)

Product

Although the basic case form asks you to specify a product name, we need specific information about which *feature* in that product is having the issue. Ideally, your report will refer to specific APIs or Cloud Console URLs (or screenshots). For APIs, you can link to the documentation page, which contains the product name in the URL.

Also tell us about the mechanism you're using to initiate the request (for example, REST API, gcloud tool, Cloud Console, or perhaps a tool like Deployment Manager. If multiple products are involved, give us each name specifically.

Examples:

- "Google Compute Engine REST API returned the following errors..."
- "The BigQuery query interface in console.cloud.google.com is hanging..."

The following statements are not specific enough to know where to look when diagnosing the issue:

- "Can't create instances..." (We need to know the method you are using to create instances.)
- "The `gcloud compute create instances` command is giving an error..."(The command syntax is incorrect, so we can't run it ourselves to reproduce the error. Also, we don't know which error you actually saw.)

Location

We need to know your data center region and zone because we often roll out changes to one region or zone at a time. The region and zone are a proxy for the version number of the underlying software. This information helps us to know if breaking changes in a particular version of our software are affecting your systems.

Examples:

- "In us-east1-a ..."

- "I tried regions us-east1 and us-central1..."

Identifiers

Specific identifiers help us identify which of your Cloud projects is experiencing the issue. We always need to know the alphanumeric project or application ID. **Project names are not helpful.** If the issue is affecting multiple projects, include every affected ID.

In addition to project or application IDs, several other identifiers help us diagnose your case:

- Instance IDs.
- BigQuery job IDs or table names.
- IP addresses.

When specifying an IP address, also tell us the context in which it is used. For example, specify if the IP connected to a Compute instance, a load balancer, a custom route, or an API endpoint. Also tell us if the IP address is not related to Google's systems (for example, if the IP address is for your home internet, a VPN endpoint, or an external monitoring system).

Examples:

- "In project robot-name-165473 or my-project-id..."
- "Across multiple projects (including my-project-id)..."
- "Connecting to GCP external IP 218.239.8.9 from our corporate gateway 56.56.56.56..."

General statements like the following are too general to help diagnose the issue:

- "One of our instances is unreachable..."
- "We can't connect from the internet..."

Useful artifacts

Providing us with artifacts related to the issue will speed up troubleshooting by helping us see exactly what you are seeing.

For example:

- Use a screenshot to show exactly what you see.

- For Web-based Interfaces, provide a .HAR (https://en.wikipedia.org/wiki/.har) (**H**ttp **AR**chive) file. The HAR analyzer (https://toolbox.googleapps.com/apps/har_analyzer/) has instructions for three major browsers.
- Attach tcpdump output, logs snippets, example stack traces.

Setting the priority and escalating

Priority helps us understand the impact this issue is having on your business, and affects how quickly we respond to resolve the issue. Priorities are defined in the following table. You can find more information at Cloud Support Procedures (https://cloud.google.com/support/docs/procedures#support_case_priority).

Priority	Definition	Example Situation
P1: Critical Impact—Service Unusable in Production	The application or infrastructure is unusable in production, having a significant rate of user-facing errors. Business impact is critical (revenue loss, potential data integrity issue, etc.).
P2: High Impact—Service Use Severely Impaired	The infrastructure is degraded in production, having a noticeable rate of user-facing errors or difficulties in spinning up a new production system. Business impact is moderate (danger of revenue loss, productivity decrease, etc.).
P3: Medium Impact—Service Use Partially Impaired	The issue is limited in scope and/or severity. The issue has no user-visible impact. Business impact is low (for example, inconvenience, minor business processes affected, etc.).
P4: Low Impact—Service Fully Usable	Little to no business or technical impact. Recommended for consultative tickets where in-depth analysis, troubleshooting or consultancy are preferred to more frequent communications.

When to set the highest priority

If you have an issue that is affecting business critical services and needs immediate attention from Google, choose "P1" as the priority. Explain to us in detail why you selected P1. Include a brief description of the impact this issue is having on your business. For example, you may consider a problem with a dev version to be P1, even if no end users are directly impacted, if it is blocking a critical security fix.

When a case is set as P1, an on duty Support team member will be immediately alerted to find the right expert to exclusively work on the issue. You will receive a quick initial response (within 15 mins for Enterprise customers and up to 1 hour for Role-Based Support Production role customers). After that, you will receive regular updates.

We appreciate detailed comments about why the prioritization increase is necessary because it helps us respond appropriately.

Response times

Issue priority levels have predefined response times defined in the Google Cloud Platform Technical Support Services Guidelines (https://cloud.google.com/terms/tssg/). If you need a response by a specific time, let us know in your report description. If a P1 issue needs to be handled around the clock, you can request "follow the sun" (https://cloud.google.com/support/docs/procedures#following_the_sun) service. These cases are re-assigned several times per day to an active support engineer.

Escalating

When circumstances change, you may need to escalate the issue to give it more prompt attention. Good reasons for escalation are:

- Increase in business impact.
- Issue needs to be resolved faster.
- Issue is "stuck" without progress after exchanging several messages.

When a GCP Support case is escalated, a Support manager is immediately notified and will update you within 1 hour or less. The Escalation Manager will own the escalation until its closure. The manager will identify and address the escalation root cause and report preventative actions to avoid similar escalations in future.

To escalate a case, you can request an escalation in the comments in the current case, or you can click the **Escalate case** button which appears 60 minutes after creating the case.

Long-running or difficult issues

Issues that take a long time to resolve can become confusing and stale. The best way to prevent this is to collect information using our Long-running issue template (https://docs.google.com/spreadsheets/d/1dpAXJ6w4RGbC91aLMUf__IpUDuehQ_yE3dXjmLGo1gE/edit#gid=0)
with the latest state summarized at the top.

To use the template, simply open the above link and make a copy. Include links to all relevant cases and internal tracking bugs. Share this doc with your account team's group and ask them to share it with specific support engineers.

This doc includes:

- A summary of the current state summarized at the top.
- A list of the hypotheses that are potentially true.
- The tests or tools that you intend to use to test each hypothesis.

Try to keep each case focused on a single issue, and avoid reopening a case to bring up a new issue.

Reporting a production outage

If the issue has caused your application to stop serving traffic to users, or has similar business-critical impact, this may be a production outage. We want to know as soon as possible. Issues that block a small number of developers, by contrast, are not what we think of as production outages.

When we get a report of a production outage, we quickly triage the situation by:

- Immediately checking for known issues affecting GCP infrastructure.
- Confirming the nature of the issue.
- Establishing communication channels.

You can expect a response with a brief message, which will contain:

- Any related known issues affecting multiple customers.
- An acknowledgement that we can observe the issue you've reported, or a request for more details.

- How we intend to communicate (for example, phone, Hangout, or case).

Therefore, it's important to quickly create a case including time, product, identifiers, and location, and then start deeper troubleshooting. Your organization may have a defined incident management process and this step should be executed very near the beginning of it.

Google's incident management process defines a key role: the incident commander. This person gets the right people involved, continually collects the latest status, and periodically summarizes the state of the issue. They delegate to others to troubleshoot and apply changes. This delegation allows us to investigate multiple hypotheses in parallel. We recommend that you establish a similar process within your organization. The person who opened the case is usually the best choice to be the incident commander because they have the most context.

Reporting a networking issue

The size and complexity of Google's network can make it difficult to identify which team owns the problem. To diagnose networking issues, we need to identify very specific root causes. Because networking error messages are often general (like, "Can't connect to server"), we need to gather detailed diagnostic information to narrow down the possible hypotheses.

Packet flow diagrams provide an excellent structure for the issue report. These diagrams describe the important hops that a packet takes along a path from source to destination, along with any significant transformations along the way.

Start by identifying the affected network endpoints by Internet IP address or by RFC 1918 (https://tools.ietf.org/html/rfc1918) private address plus an identifier for the network. For example, 2.3.4.5 or 10.2.3.4 on the Compute Engine project's default network.

Note anything meaningful about the endpoints, such as:

- Who controls them.
- Whether they are associated with a DNS hostname.
- Any intermediate encapsulation and/or indirection, such as VPN tunneling, proxies, and NAT gateways.
- Any intermediate filtering, like firewalls or CDN or WAF.

Many problems that manifest as high latency or intermittent packet loss will require a path analysis and/or a packet capture for diagnosis.

- **Path analysis** is a list of all hops that packets traverse and is familiar as "traceroute". We often use MTR (https://en.wikipedia.org/wiki/MTR_(software)) and/or tcptraceroute because they have better diagnostic power, so please be familiar with these tools.
- **Packet capture** (aka "pcap," from the name of the library "libpcap") is an observation of real network traffic. It's important to take a packet capture for both endpoints, at the same time, which can be tricky. It's a good idea to practice with the necessary tools (for example tcpdump (https://en.wikipedia.org/wiki/Tcpdump) or Wireshark (https://en.wikipedia.org/wiki/Wireshark)) and make sure they are installed before you need them.

Sample Cases

Example 1

JobName:

A_ATL_BIG1toBQ_big_04)201704202

00045_491

Source:

S3_avl-transfer

Destination:

CloudStorage: avl-transfer

Start time (ISO 8601 format): 2017-04-20 20:14:43 PDT

End time (ISO 8601 format): 2017-04-21 at 10:03:44 PDT

I started a file transfer at 2017-04-20 at 20:14:43 PDT using the transfer API. This job normally takes 10 minutes to complete, but in this case the job was still running when I canceled it the next day (2017-04-21 at 10:03:44 PDT). This is not an isolated event; several other jobs involving the transfer API had intermittent, significant delays.

Please investigate the cause of the delays and advise of any best practices we can implement to prevent these issues in the future.

Example 2

Start time (ISO 8601 format): 2017-05-12 at 11:03:43

End time (ISO 8601 format): The issue is still happening as of the time of this report.

Issue summary:

`/cron/payments-service/sync-v2-batch` cron using the App Engine Task Queue API has stopped running since 2017-05-12 at 11:03:43. We rely on this job to handle payments correctly.

We saw datastore and queue errors and then the cron stopped running. We attempted unsuccessfully to fix the issue by re-uploading cron.xml. Here is the error trace:

`[error trace]`

Please advise if the issue is with the API or our implementation and let us know next steps.
