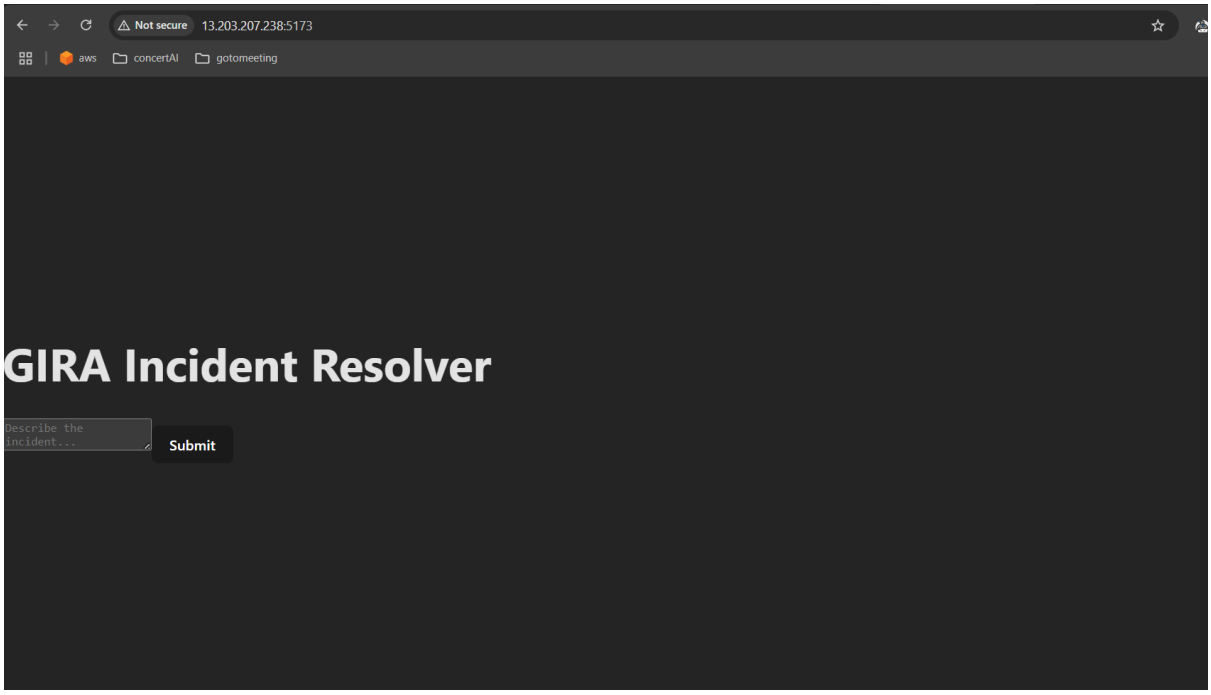
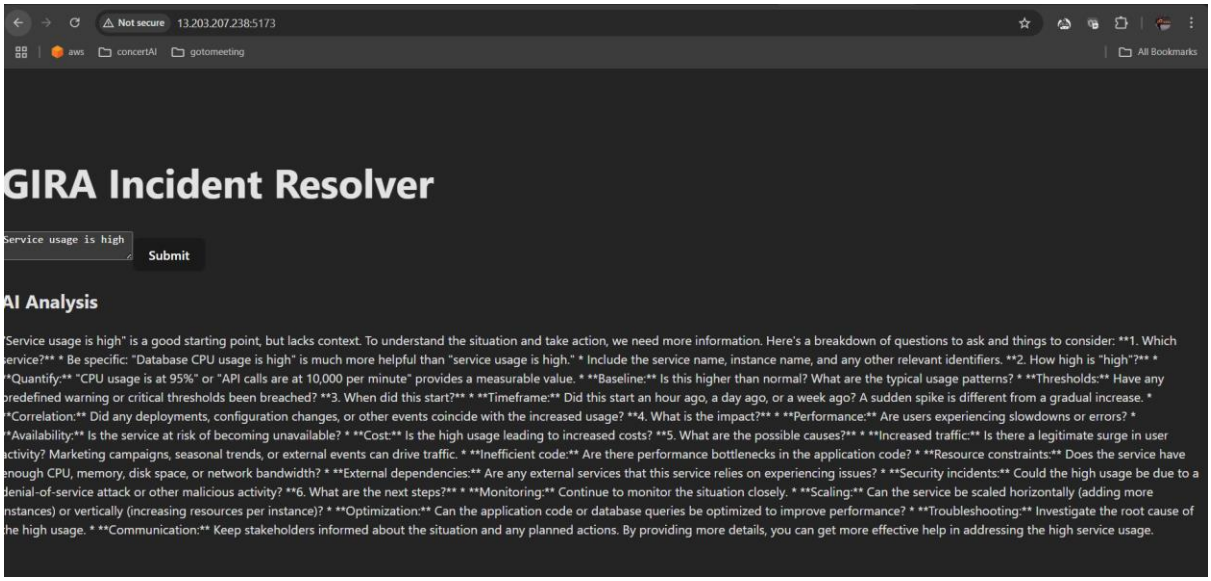

Not secure
13.203.207.238:5173
aws
concertAI
gotomeeting
GIRA Incident Resolver
Describe the incident...
Submit


Not secure
13.203.207.238:5173
aws
concertAI
gotomeeting
All Bookmarks
GIRA Incident Resolver
Service usage is high
Submit
AI Analysis
"Service usage is high" is a good starting point, but lacks context. To understand the situation and take action, we need more information. Here's a breakdown of questions to ask and things to consider: **1. Which service?** * Be specific: "Database CPU usage is high" is much more helpful than "service usage is high." * Include the service name, instance name, and any other relevant identifiers. **2. How high is "high"?** * **Quantify:** "CPU usage is at 95%" or "API calls are at 10,000 per minute" provides a measurable value. * **Baseline:** Is this higher than normal? What are the typical usage patterns? * **Thresholds:** Have any predefined warning or critical thresholds been breached? **3. When did this start?** * **Timeframe:** Did this start an hour ago, a day ago, or a week ago? A sudden spike is different from a gradual increase. * **Correlation:** Did any deployments, configuration changes, or other events coincide with the increased usage? **4. What is the impact?** * **Performance:** Are users experiencing slowdowns or errors? * **Availability:** Is the service at risk of becoming unavailable? * **Cost:** Is the high usage leading to increased costs? **5. What are the possible causes?** * **Increased traffic:** Is there a legitimate surge in user activity? Marketing campaigns, seasonal trends, or external events can drive traffic. * **Inefficient code:** Are there performance bottlenecks in the application code? * **Resource constraints:** Does the service have enough CPU, memory, disk space, or network bandwidth? * **External dependencies:** Are any external services that this service relies on experiencing issues? * **Security incidents:** Could the high usage be due to a denial-of-service attack or other malicious activity? **6. What are the next steps?** * **Monitoring:** Continue to monitor the situation closely. * **Scaling:** Can the service be scaled horizontally (adding more instances) or vertically (increasing resources per instance)? * **Optimization:** Can the application code or database queries be optimized to improve performance? * **Troubleshooting:** Investigate the root cause of the high usage. * **Communication:** Keep stakeholders informed about the situation and any planned actions. By providing more details, you can get more effective help in addressing the high service usage.


Not secure
13.203.207.238:5173
aws
concertAI
gotomeeting
All Bookmarks
GIRA Incident Resolver
service apache is not running
Submit
AI Analysis
The message "service apache is not running" indicates that the Apache web server is not currently active on your system. Here's how to troubleshoot and fix this: **1. Verify the Service Name:** * **System-Specific Commands:** The command `service apache` might not be the correct one for your system. * **Debian/Ubuntu (systemd):** Use `systemctl status apache2` * **CentOS/RHEL 7+ (systemd):** Use `systemctl status httpd` * **Older CentOS/RHEL (SysVinit):** Use `service httpd status` or `/etc/init.d/httpd status` * **macOS (Homebrew):** Use `brew services list` to see if it's running and `brew services start httpd` to start it. **2. Start the Apache Service:** Once you've determined the correct service name, use the appropriate command to start it: * **Debian/Ubuntu (systemd):** `sudo systemctl start apache2` * **CentOS/RHEL 7+ (systemd):** `sudo systemctl start httpd` * **Older CentOS/RHEL (SysVinit):** `sudo service httpd start` or `sudo /etc/init.d/httpd start` * **macOS (Homebrew):** `brew services start httpd` **3. Check for Errors:** If the service fails to start, check the logs for error messages. Common locations include: * **Debian/Ubuntu:** `/var/log/apache2/error.log` * **CentOS/RHEL:** `/var/log/httpd/error_log` or `/var/log/httpd/error.log` * **macOS (Homebrew):** `/usr/local/var/log/httpd/error_log` **4. Common Issues and Solutions:** * **Port Conflicts:** Another application might be using port 80 (the default for HTTP) or 443 (for HTTPS). Check if anything else is using these ports. You can use `netstat -tulpn` or `ss -tulpn` to see what's listening. * **Configuration Errors:** A syntax error in the Apache configuration files (usually `httpd.conf` or `apache2.conf` and related files in the `conf.d` or `sites-available` directories) can prevent Apache from starting. Carefully review the logs for specific lines and files mentioned. * **Missing Dependencies:** Apache might depend on other libraries or modules. Ensure these are installed. * **Firewall Issues:** If you have a firewall enabled, make sure it allows traffic on ports 80 and 443. Use `sudo ufw allow 80` and `sudo ufw allow 443` for UFW (Ubuntu Firewall), or `sudo firewall-cmd --permanent --add-service=http` and `sudo firewall-cmd --permanent --add-service=https` followed by `sudo firewall-cmd --reload` for Firewalld (CentOS/RHEL). * **SELinux (CentOS/RHEL):** If SELinux is enforcing, it might be blocking Apache. Check the SELinux logs (`/var/log/audit/audit.log`) and use `sudo setenforce 0` (temporarily disable SELinux) to test if it's the cause. If it is, you'll need to configure SELinux to allow Apache to run. **5. Enable Apache to Start on Boot:** After successfully starting Apache, enable it to start automatically on system boot: * **Debian/Ubuntu (systemd):** `sudo systemctl enable apache2` * **CentOS/RHEL 7+ (systemd):** `sudo systemctl enable httpd` * **Older CentOS/RHEL (SysVinit):** `sudo chkconfig httpd on` * **macOS (Homebrew):** `brew services run httpd` (this command will keep it running even after reboot) By systematically checking these steps, you should be able to identify the reason why Apache isn't running and get it back online. Remember to replace `apache2` or `httpd` with the correct service name for your system. If you encounter specific errors, providing the error messages will help in getting more tailored assistance.