1. Instance count >= 2

   Having an instance count greater than or equal to 2 in a cloud computing environment, such as in the context of services like Cloud Run, often indicates that multiple instances of your application are running simultaneously. This can happen for various reasons, and resolving it may depend on the specific cloud service and configuration. Here are some common conditions and potential solutions:

   Auto-scaling Configuration: If your service is configured to auto-scale based on traffic, multiple instances may be created to handle increased demand. To resolve this, you can adjust the auto-scaling settings to better match your application's traffic patterns. This may involve changing parameters such as minimum and maximum instance counts, cooldown periods, or scaling thresholds.

   Manual Scaling: If you have manually configured your service to run a specific number of instances, the instance count will remain constant unless manually adjusted. Review your manual scaling configuration and update it based on your current requirements.

   Persistent Connections or Sessions: If your application relies on persistent connections or sessions and instances are being created to handle new connections, you might want to evaluate whether the application design can be optimized to reduce the need for multiple instances. Using stateless architectures or externalizing session management could be potential solutions.

   High Request Rate: A high rate of incoming requests may trigger the creation of additional instances. You can optimize your application code, use caching, or implement other performance improvements to handle more requests with fewer instances.

   Resource Constraints: If your instances are being created due to resource constraints (CPU, memory), consider reviewing your resource allocation settings. Adjust the allocated resources per instance to better match your application's requirements.

   Startup Latency: Instances may be created to handle spikes in traffic or to replace instances experiencing cold starts. Optimize your application's startup

time by implementing warm-up requests or other strategies to reduce latency during instance initialization.

Dependency Scaling: If your application has dependencies that scale independently, ensure that those dependencies are also configured appropriately. For example, if your application relies on a database, the database connection pool settings may impact instance count.

To resolve the issue of having an instance count greater than or equal to 2:

Review and adjust your auto-scaling settings if applicable.
Check if there are any manual scaling configurations that need adjustment.
Optimize your application code, especially in terms of handling persistent connections and sessions.
Monitor and optimize resource usage to prevent unnecessary instance creation.
Implement strategies to reduce startup latency, such as warm-up requests.

By addressing these factors, you can better control the instance count and ensure that your cloud service is efficiently handling the workload.