A micro service based application deployed in Kubernetes is experiencing random 504 errors. How would you identify and resolve the root cause?

The 504 (Gateway Timeout) status code indicates that the server, while acting as a gateway or proxy, did not receive a timely response from an **upstream server it needed to access in** order to complete the request.

To identify and resolve **504 Gateway Timeout** errors in a **Kubernetes-based microservices** application, you typically look for the following issues:

Common Root Causes

1. **Downstream service is slow or unresponsive** (long processing time).
2. **Readiness probe failures** or **pods not available**.
3. **Service misconfiguration** (timeouts not matching between client & backend).
4. **Network issues** or **load balancer timeouts** (Nginx/Ingress).
5. **Resource constraints** (CPU/Memory throttling causing delays).

User => FrontEnd =>Backend

We'll simulate:

- A **frontend service** that calls a **backend service**.
- The backend will deliberately **delay response** to simulate slowness.

What Happens:

- When you access the `gateway` service, it tries to call `slow-api`.
- `slow-api` delays for **10 seconds**, but `gateway` times out in **5 seconds**.
- This causes a **504 Gateway Timeout** response.

```
kubectl apply -f gateway.yaml

kubectl port-forward svc/gateway 8080:80

curl http://localhost:8080
```

How to fix ?

Change `setTimeout` in `slow-api` to `2000`	Gateway returns success
Change gateway timeout to `20000`	It waits longer before 504
Add readiness probe to `slow-api`	K8s waits before traffic

How to Debug Such Errors

1. **Check logs** of gateway and backend pods:
   ```
   kubectl logs gateway
   kubectl logs slow-api
   ```
2. **Increase timeout settings** if backend legitimately takes time.
3. **Use liveness/readiness probes** to ensure pods are healthy.
4. **Monitor Ingress (like NGINX) timeouts**:
   - Check `proxy_read_timeout` and `proxy_connect_timeout`.
5. **Set resource limits properly** to avoid throttling:

```
yaml

resources:
  requests:
    cpu: "100m"
    memory: "128Mi"
  limits:
    cpu: "500m"
```

```
    memory: "256Mi"
```

A microservice-based application deployed in Kubernetes is intermittently experiencing **504 Gateway Timeout** errors. Which of the following is **NOT** a likely cause of this issue?

A. The downstream service takes too long to respond due to heavy processing.
B. The upstream service returns a 200 OK status code too quickly.
C. The readiness probe fails, making pods unavailable temporarily.
D. The timeout settings between the gateway and backend services are mismatched.
E. The pod is being throttled due to CPU or memory limits.

✅ Correct Answer:

B. The upstream service returns a 200 OK status code too quickly.
