PRACTICAL TEST 1: Creating a Virtual Machine on Google Cloud

Aim: To create and configure a Virtual Machine (VM) ensuring high availability, low latency, and scalability.

System Requirements:
• Google Cloud account with billing enabled
• Access to Compute Engine
• Web browser and stable internet

Procedure:
1. Console → Compute Engine → VM Instances → Create Instance.
2. Choose Machine Type: e2-medium (2 vCPU, 4GB RAM).
3. Select OS Image: Debian/Ubuntu LTS.
4. Region & Zone: us-central1-a for reliability.
5. Enable Firewall rules: Allow HTTP (80) & HTTPS (443).
6. Enable Automatic Restart under Availability Policy.
7. Click Create.

Flow Chart:
Start → Choose Machine Type → Select OS → Configure Region/Zone → Allow Firewall → Enable Auto Restart → Create VM → End

Result: VM created successfully with HTTP/HTTPS access and high availability.

PRACTICAL TEST 2: Deploying VM Instances with Load Balancer and Autoscaling

Aim: To deploy multiple identical VMs behind a load balancer with autoscaling.

System Requirements:
- Google Cloud project with Compute Engine enabled
- IAM access for Instance Groups and Load Balancer

Procedure:

1. Create Instance Template:

```
gcloud compute instance-templates create web-template --machine-type=e2-medium
--image-family=debian-11 --image-project=debian-cloud
```

2. Create Managed Instance Group (MIG):

```
gcloud compute instance-groups managed create web-group --template=web-template --size=2
--zone=us-central1-a
```

3. Configure Autoscaling (CPU 60%):

```
gcloud compute instance-groups managed set-autoscaling web-group --max-num-replicas=5
--target-cpu-utilization=0.6
```

4. Create HTTP Load Balancer and attach MIG.

Flow Chart:

Start → Create Template → Create MIG → Enable Autoscaling → Configure Load Balancer → End

Result: Autoscaling VM cluster deployed with load balancing for efficient traffic handling.

PRACTICAL TEST 3: Creating Secure Cloud Storage Bucket

Aim: To create a Cloud Storage bucket for storing images with restricted access.

System Requirements:
• Google Cloud Storage and IAM permissions enabled

Procedure:
Console Steps:
1. Console → Cloud Storage → Create Bucket.
2. Name: app-image-bucket, Region: us-central1.
3. Storage Class: Standard, disable Public Access.
4. Add IAM: Grant Storage Object Admin to web app service account only.

CLI Steps:
```
gcloud storage buckets create gs://app-image-bucket --location=us-central1
--uniform-bucket-level-access
gsutil iam set 'serviceAccount:app-sa@project-id.iam.gserviceaccount.com:roles/storage.objectAdmin'
gs://app-image-bucket
```

Flow Chart:
Start → Create Bucket → Disable Public Access → Add IAM → Verify Access → End

Result: Secure private bucket created with restricted read/write permissions.

PRACTICAL TEST 4: Setting up a Cloud SQL MySQL Instance

Aim: To deploy a production-grade MySQL instance and configure secure app access.

System Requirements:
• Cloud SQL API enabled
• App server IP or VPC available for connection

Procedure:
1. Console → SQL → Create Instance (MySQL 8.0).
2. Choose db-custom-2-4096, 20GB SSD storage.
3. Authorize network (app server IP or private VPC).
4. Create DB and user:

```
gcloud sql databases create appdb --instance=mysql-prod
gcloud sql users create appuser --instance=mysql-prod --password=StrongPwd123
```

5. Store credentials in Secret Manager:

```
gcloud secrets create db-password --replication-policy=automatic
echo -n "StrongPwd123" | gcloud secrets versions add db-password --data-file=-
```

Flow Chart:
Start → Create SQL Instance → Configure Network → Create DB/User → Secure Credentials → End

Result: MySQL database deployed securely with credentials stored in Secret Manager.

PRACTICAL TEST 5: Creating a Private VPC Network with No Internet Access

Aim: To design and implement a VPC for internal-only communication without external Internet access.

System Requirements:
• Google Cloud SDK or Console access
• Compute Admin / Network Admin IAM roles

Procedure:
1. Create Custom VPC & Subnet:
```
gcloud compute networks create private-vpc --subnet-mode=custom
gcloud compute networks subnets create private-subnet --network=private-vpc --region=us-central1
--range=10.10.0.0/24 --enable-private-ip-google-access
```
2. Add Firewall Rules:
Allow internal:
```
gcloud compute firewall-rules create allow-internal --network=private-vpc
--allow=tcp,udp,icmp --source-ranges=10.10.0.0/16
```
Deny external:
```
gcloud compute firewall-rules create deny-internet --network=private-vpc
--direction=EGRESS --action=DENY --rules=all --destination-ranges=0.0.0.0/0
```
3. Create VM with no external IP:
```
gcloud compute instances create internal-vm --subnet=private-subnet --no-address
--zone=us-central1-a
```
4. Verify: Ping internal VM, external ping blocked.

Flow Chart:
Start → Create VPC → Create Subnet → Add Firewall → Deploy VM (no external IP) → Test → End

Result: A private-only VPC created, internal communication enabled, external access blocked securely.