**3. Container Lifecycle Management**

b. Write a document detailing the container lifecycle, including stages such as creation, starting, stopping, and removing containers.

c. Provide commands for each stage, demonstrating how to manage containers on AWS, Azure, and GCP.

**Container Lifecycle Management**

**1. Container Lifecycle Stages**

**a. Creation**

The first step in container management is creating a container from an image. This involves downloading the necessary image from a container registry and setting up the environment.

**b. Starting**

Once a container is created, it needs to be started to run the application. A container can be started immediately upon creation or restarted later.

**c. Stopping**

Stopping a container gracefully terminates the running processes inside it. This ensures that data is saved correctly and resources are freed.

**d. Removing**

When a container is no longer needed, it should be removed to free up storage and resources.

**2. Managing Containers on AWS, Azure, and GCP**

**AWS (Amazon Web Services)**

AWS provides container management through services like Amazon ECS (Elastic Container Service) and Amazon EKS (Elastic Kubernetes Service).

**a. Creating a Container**

aws ecs create-cluster --cluster-name my-cluster

aws ecs create-service --cluster my-cluster --service-name my-service --task-definition my-task

**b. Starting a Container**

aws ecs update-service --cluster my-cluster --service my-service --desired-count 1

**c. Stopping a Container**

aws ecs update-service --cluster my-cluster --service my-service --desired-count 0

**d. Removing a Container**

aws ecs delete-service --cluster my-cluster --service my-service --force

**Azure (Microsoft Azure)**

Azure provides container management through Azure Container Instances (ACI) and Azure Kubernetes Service (AKS).

**a. Creating a Container**

az container create --resource-group myResourceGroup --name myContainer --image myImage --dns-name-label mycontainer --ports 80

**b. Starting a Container**

Azure containers start upon creation, but if stopped:

az container start --resource-group myResourceGroup --name myContainer

**c. Stopping a Container**

az container stop --resource-group myResourceGroup --name myContainer

**d. Removing a Container**

az container delete --resource-group myResourceGroup --name myContainer --yes

**GCP (Google Cloud Platform)**

GCP provides container management through Google Kubernetes Engine (GKE) and Cloud Run.

**a. Creating a Container**

gcloud run deploy my-service --image gcr.io/my-project/my-image --platform managed

**b. Starting a Container**

gcloud run services update-traffic my-service --to-latest

**c. Stopping a Container**

gcloud run services update-traffic my-service --to-zero

**d. Removing a Container**

gcloud run services delete my-service

**Conclusion**

Each cloud provider offers tools and commands to manage the lifecycle of containers efficiently. By following these commands, users can create, start, stop, and remove containers across AWS, Azure, and GCP environments effectively.