Kubernetes:

1. What is replication controller and ReplicaSet?
2. How many types of Services we have in Kubernetes, name them?
3. What is the difference between pod and container?
4. What is the difference between virtualization and Containerization?

Terraform:

1. What does the terraform init do?
2. What is the difference between plan and apply?
3. How do we secure terraform state file?
4. How do you define local variables in terraform?

Azure:

1. What is difference between api gateway and app gateway?
2. What is the difference between service endpoint, private end point and private link?
3. What are the storage types available in azure?
4. what is CDN why do we use that?

Pipelines:

1. what CI/CD pipelines you worked on, explain the stages Involved?

Docker:

1. what is docker prune?
2. Difference between docker add and copy?

Motivity lab:

**Azure DevOps:**

What is deployment group?

Explain Azure DevOps Pipeline like stages, build, release pipeline?

Explain how to release dev to stage to production?

**Kubernetes:**

1. How many services there in k8s ?
2. What is ingress controller in k8s?
3. How to pull image from private cluster in k8s? 1. service connection establishing

**Docker:**

What is docker file?

What is different b/w add and copy?

What is different b/w cmd and entrypoint

What is difference b/w env and argument?

**Terraform**:

1.What is difference b/w provider and provisioner?

**Provisioner:**

1. **Definition:**
   * A provisioner in Terraform is responsible for executing scripts or commands on a local machine or a remote resource after a resource is created. Provisioners are used to perform tasks such as configuration management, initialization, or custom actions.
2. **Examples:**
   * Common Terraform provisioners include:
     + **local-exec** for executing commands on the local machine where Terraform is run.
     + **remote-exec** for executing commands on a remote resource (e.g., a virtual machine)

2. How to create resource in multiple instances using terraform?

In Terraform, you can create multiple instances of a resource using the **count** parameter or by using the **for\_each** expression, depending on your specific use case.

Let me explain both approaches:

Using “Count”

**resource "aws\_instance" "example" {**

**count = 3**

**ami = "ami-0c55b159cbfafe1f0"**

**instance\_type = "t2.micro"**

**}**

**In this example, three instances of the aws\_instance resource will be created. The instances will be identical, and the count parameter specifies how many instances to create.**

**The instances will be named**

**aws\_instance.example[0],**

**aws\_instance.example[1],**

**and aws\_instance.example[2].**

**Using for\_each:**

variable "instance\_names" {

type = list(string)

default = ["instance-1", "instance-2", "instance-3"]

}

resource "aws\_instance" "example" {

for\_each = toset(var.instance\_names)

ami = "ami-0c55b159cbfafe1f0"

instance\_type = "t2.micro"

tags = {

Name = each.value

}

}

**In this example, you use the for\_each expression with a set of instance names provided through a variable. This approach allows you to create instances with different configurations. The instances will be named according to the values in the instance\_names variable.**

**Choose the approach that best fits your requirements.**

**If you need identical instances, use count.**

**If you need instances with different configurations or names, use for\_each.**

**Additionally, note that for\_each is more flexible and powerful, but it requires Terraform 0.12.6 or later. If you're using an older version, you might need to upgrade to take advantage of for\_each**