Cyber software technology:

1.Tell me about your self and current project and roles and responsibility regarding devops

2. How to monitoring pod in Kubernetes

Ans: kubectl get pods

kubectl describe pod <pod-name>

3.How to delete pod

1. **Delete a Pod by Name**:

kubectl delete pod <pod-name>

2. **Delete Pods Using Labels**:

kubectl delete pods -l <label-key>=<label-value>

3. **Delete Pods in a Namespace**

**kubectl delete pod -n <namespace> <pod-name>**

4. Delete All Pods in a Namespace

kubectl delete pods --all -n <namespace>

4. how to restore pods in k8s

**Scale the Deployment/ReplicaSet**:

If you want to ensure that a certain number of pods are always running, you can scale up the deployment or replica set. This will create new pods to replace any that have been deleted

kubectl scale deployment <deployment-name> --replicas=<desired-replica-count>

5. how to execute or build docker image

docker build -t my-custom-image .

**Build the Docker Image**:

docker build -t my-node-app .

**Run a Container from the Image**

docker run -p 4000:3000 my-node-app

Detached run

docker run -d -p 4000:3000 my-node-app

**Key Differences**:

* **git fetch** only downloads changes from the remote repository but does not automatically integrate them into your local branch.
* **git pull** is a combination of **git fetch** followed by an automatic merge, potentially leading to merge conflicts.
* **git fetch** is typically used when you want to review changes before integrating them or when you want to update your local repository without affecting your current branch.
* **git pull** is used when you want to quickly update your local branch with the latest changes from the remote.

What is azure Artifacts?

Azure Artifacts enables developers to share their code efficiently and manage all their packages from one place. With Azure Artifacts, developers can publish packages to their feeds and share it within the same team, across organizations, and even publicly.

1. **Centralized Package Management**:
   * Azure Artifacts provides a centralized location to store and manage all your organization's software artifacts and packages.
2. **Support for Multiple Package Types**:
   * It supports various package formats, including NuGet, npm, Maven, and Python packages. This allows you to manage dependencies for a wide range of projects and applications.
3. **Private Feeds for Secure Package Storage**:
   * You can create private feeds to store sensitive or proprietary packages securely within your organization. This ensures that only authorized team members can access and use these packages.

Variables:

Scope: Variables are defined at the pipeline level. They are specific to a single pipeline and can be used within that pipeline and its stages, jobs, and tasks.

Syntax :

variables:

myVariable: 'someValue'

Variable Groups:

Scope: Variable groups are defined at the project or organization level. They can be used across multiple pipelines within the same project or across projects in the organization

Syntax:

variables:

- group: 'MyVariableGroup'

Refresh

Rm