**Kubernetes Tasks**

**Task 1:**

Deploy Pods in Kubernetes Cluster: -

The Nautilus DevOps team is diving into Kubernetes for application management. One team member has a task to create a pod according to the details below:

1. Create a pod named pod-nginx using the nginx image with the latest tag. Ensure to specify the tag as nginx:latest.
2. Set the app label to nginx\_app, and name the container as nginx-container.

Note: The kubectl utility on jump\_host is configured to operate with the Kubernetes cluster.

**Task 2:**

Deploy Applications with Kubernetes Deployments: -

The Nautilus DevOps team is delving into Kubernetes for app management. One team member needs to create a deployment following these details:

Create a deployment named nginx to deploy the application nginx using the image nginx:latest (ensure to specify the tag)

Note: The kubectl utility on jump\_host is set up to interact with the Kubernetes cluster.

**Task 3:**

Setup Kubernetes Namespaces and PODs: -

The Nautilus DevOps team is planning to deploy some micro services on Kubernetes platform. The team has already set up a Kubernetes cluster and now they want to set up some namespaces, deployments etc. Based on the current requirements, the team has shared some details as below:

Create a namespace named dev and deploy a POD within it. Name the pod dev-nginx-pod and use the nginx image with the latest tag. Ensure to specify the tag as nginx:latest.

Note: The kubectl utility on jump\_host is configured to operate with the Kubernetes cluster.

Task 4:

Set Resource Limits in Kubernetes Pods: -

An application currently running on the Kubernetes cluster employs the nginx web server. The Nautilus application development team has introduced some recent changes that need deployment. They've crafted an image nginx:1.17 with the latest updates.

Execute a rolling update for this application, integrating the nginx:1.17 image.

The deployment is named nginx-deployment. Ensure all pods are operational post-update.

Task 5:

Execute Rolling Updates in Kubernetes: -

An application currently running on the Kubernetes cluster employs the nginx web server. The Nautilus application development team has introduced some recent changes that need deployment. They've crafted an image nginx:1.17 with the latest updates. Execute a rolling update for this application, integrating the nginx:1.17 image. The deployment is named nginx-deployment. Ensure all pods are operational post-update.

Task 6:

Revert Deployment to Previous Version in Kubernetes

Earlier today, the Nautilus DevOps team deployed a new release for an application. However, a customer has reported a bug related to this recent release. Consequently, the team aims to revert to the previous version.

There exists a deployment named nginx-deployment; initiate a rollback to the previous revision.

Task 7:

**Schedule Cronjobs in Kubernetes**

The Nautilus DevOps team is setting up recurring tasks on different schedules. Currently, they're developing scripts to be executed periodically. To kickstart the process, they're creating cron jobs in the Kubernetes cluster with placeholder commands. Follow the instructions below:

1. Create a cronjob named xfusion.
2. Set Its schedule to something like \*/5 \* \* \* \*. You can set any schedule for now.
3. Name the container cron-xfusion.
4. Utilize the nginx image with latest tag (specify as nginx:latest).
5. Execute the dummy command echo Welcome to xfusioncorp!.
6. Ensure the restart policy is OnFailure.