Argo Evetn

1. Create SSH Keys:

- If you don't have SSH keys available, follow a guide to create them. SSH keys are used for secure communication between systems, like fetching data from a Git repository.

2. Create K8s Secret for SSH Keys:

- Use the `kubectl` command to create a Kubernetes secret named `git-ssh` in the `argo-events` namespace. This secret will hold your SSH keys.

- The command syntax is:

```
    kubectl -n argo-events create secret generic git-ssh --from-file=key=.ssh/<YOUR_SSH_KEY_FILE_NAME>
```

- Replace `<YOUR_SSH_KEY_FILE_NAME>` with the actual name of your SSH key file.

3. Create K8s Secret for Known Hosts:

- Similarly, create another Kubernetes secret named `git-known-hosts` in the `argo-events` namespace. This secret will hold your known hosts file, which helps verify server identities.

- The command syntax is:

```
    kubectl -n argo-events create secret generic git-known-hosts --from-file=ssh_known_hosts=.ssh/known_hosts
```

4. Create a Sensor for Git Trigger:

- Create a sensor in Kubernetes that will listen for events from your Git repository. This sensor is configured to trigger workflows based on changes in the Git repository.

- Use the following command to create the sensor:

```
    kubectl -n argo-events apply -f https://raw.githubusercontent.com/argoproj/argo-events/stable/examples/tutorials/03-trigger-sources/sensor-git.yaml
```

5. Send a POST Request:

- Use either Curl or Postman to send a POST request to `http://localhost:12000/example`. This request simulates an event that triggers the sensor.

- Here's an example Curl command:

```
    curl -d '{"message":"ok"}' -H "Content-Type: application/json" -X POST http://localhost:12000/example
```

6. Check Argo Workflows:

- After sending the POST request, check if an Argo workflow is created. The sensor, when triggered, should start the workflow defined in your Argo Git project.