**Budegt-Manager Jenkins Integration:**

**Step 1: Jenkins Setup on macOS**

1. **Install Jenkins**:
   * The command brew install jenkins-lts installs Jenkins Long-Term Support version via Homebrew. This is correct.
   * Then, brew services start jenkins-lts starts Jenkins as a service, so it will run continuously even after you reboot your system.
   * **Access Jenkins UI**: By default, Jenkins will be available at http://localhost:8080, and that’s correct.

**Install docker and Kubernetes in Jenkins server(in my case I have used Docker-desktop which provides both)**

**Step 2: Configure Docker and Kubernetes in Jenkins**

1. **Install Necessary Plugins**:
   * You should install the **Docker Pipeline**, **Kubernetes**, and **Kubernetes CLI Plugin** through **Manage Jenkins > Manage Plugins**.
   * This will enable Jenkins to interact with Docker and Kubernetes. This part is also correct.
2. **Configure Docker in Jenkins**:
   * **Global Tool Configuration**:
     + Go to **Manage Jenkins > Global Tool Configuration**.
     + Add Docker installation by clicking **Add Docker**.
     + For **Docker executable**, point it to /usr/local/bin/docker (as verified by which docker).
     + This ensures Jenkins knows where Docker is installed.
   * **Set the PATH Variable**:
     + Go to **Manage Jenkins > Configure System**.
     + In the **Global Properties** section, check the **Environment Variables** box.

**Add variable: Name: PATH**

**Value: /bin:/usr/bin:/usr/local/bin**

The PATH variable tells the operating system where to look for executables. If docker or sh is installed in directories like /usr/local/bin, but that path is not included in the Jenkins PATH, Jenkins won't be able to find the Docker executable and will give the "command not found" error.

1. **Configure Kubernetes in Jenkins**:
   * **Add Kubernetes Cloud**:
     + Go to **Manage Jenkins > Configure System**.
     + Under the **Clouds** section, click **Add Kubernetes**.
     + In the **Kubernetes URL** field, leave it blank (it will auto-detect when using Docker Desktop, which is common on macOS).
     + Add Kubernetes credentials to allow Jenkins to interact with your Kubernetes cluster.
   * **Add Kubernetes Credentials**:
     + You’ll need to retrieve the kubeconfig content using the kubectl config view --raw command (as you mentioned).
     + Paste this content into the **Secret Text** in **Manage Jenkins > Configure System** under the **Kubernetes** credentials.
     + Then, in the dropdown, select the created credentials.

**Step 3: Create a Pipeline**

* **Create a Pipeline Job**:
  + Now, you’re ready to create a Jenkins pipeline job.
  + In the pipeline configuration, choose **Pipeline script from SCM**.
  + Provide the correct GitHub repository URL and select the appropriate branch that contains your Jenkinsfile.
  + Use **GitHub credentials** if necessary to access the private repository.