Deployment Documentation

1. Prepare the Infrastructure:

- Set up three VMs/VPS instances. You can use cloud providers like Google Cloud, AWS, or Azure, or local virtualization tools like VirtualBox or VMware.
- Install the desired operating system on each VM.

2. Install Kubernetes:

- Follow the Kubernetes documentation or use tools like kubeadm, kops, or k3s to set up a Kubernetes cluster on the VMs.

3. Deploy the Application:

- Choose a sample application or service that you want to deploy. For example, you can use the Docker Swarm microservice demo you mentioned earlier.
- Create Kubernetes deployment manifests for the chosen service.

4. Set Up CI/CD with GitLab:

- Install and configure GitLab on a separate server or use GitLab.com.
- Create a new GitLab repository for your Kubernetes project.
- Configure a `.gitlab-ci.yml` file to define the CI/CD pipeline stages, including steps for testing, building, and deploying your Kubernetes application.
- Add CI/CD variables for sensitive information (e.g., API keys, secrets).

5. Demonstrate CI/CD Features:

- Commit and push changes to your GitLab repository.
- Observe the GitLab CI/CD pipeline running automatically.
- Check the pipeline logs, artifacts, and deployment status.
- Experiment with triggering manual pipeline runs.
- Test rollback functionality (if implemented in your CI/CD pipeline).

6. Set Up Logging Service (Using GCP Stackdriver):

- Navigate to the Google Cloud Console.
- Enable the Stackdriver Logging API.
- Integrate Stackdriver Logging with your Kubernetes cluster by installing the Stackdriver Kubernetes Engine Monitoring agent. You can follow Google's documentation for instructions.

7. Integrate Logging with CI/CD:

- Enhance your CI/CD pipeline to include steps for sending logs to Stackdriver Logging.
- Use the Google Cloud Build service to automatically build and test your application. Configure it to push build logs to Stackdriver.

8. Set Up Monitoring Service (Using GCP Stackdriver Monitoring):

- Enable the Stackdriver Monitoring API in the Google Cloud Console.
- Integrate Stackdriver Monitoring with your Kubernetes cluster by installing the Stackdriver Kubernetes Engine Monitoring agent. Follow Google's documentation for instructions.

9. Integrate Monitoring with CI/CD:

- Enhance your CI/CD pipeline to include steps for sending performance metrics to Stackdriver Monitoring.
- Leverage the Stackdriver Monitoring dashboard to visualize and analyze performance metrics, such as response times, error rates, and resource utilization.

10. Demonstrate Logging and Monitoring:

- Generate logs intentionally or let the application create logs during normal usage.
- Access logs through the Stackdriver Logging interface in the Google Cloud Console.
- Explore monitoring and log visualization features provided by Stackdriver Monitoring.

11. Document Observations and Recommendations:

- Document any challenges faced during the deployment process.
- Provide recommendations for optimizations or improvements.
- Include troubleshooting steps for common issues.

12. Conclusion:

- Summarize the deployment process.
- Highlight key features of the CI/CD pipeline and logging setup.
- Mention any future improvements or additional features.

Remember to replace placeholder values in the documentation with your specific project details. This documentation provides a high-level overview, and you can expand on each section as needed.