



PROJECT SPECIFICATION

Give Your Application Auto-Deploy Superpowers

Section 1: Selling CI/CD to your Team/Organization

CRITERIA	MEETS SPECIFICATIONS
Explain the fundamentals and benefits of CI/CD to achieve, build, and deploy automation for cloud-based software products.	The CI/CD benefits proposal contains essential benefits of CI/CD, and describes the business context that will benefit from the automation tools. Explanation should include benefits that translate to revenue and cost for the business.

Section 2: Deploying Working, Trustworthy Software

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Utilize Deployment Strategies to design and build CI/CD pipelines that support Continuous Delivery processes.	<p>A public git repository with your project code. [URL01]</p> <p>Evidence of code-based CI/CD configuration in the form of yaml files in your git repository.</p> <p>Console output of various pre-deploy job failure scenarios:</p> <ul style="list-style-type: none">• Build Jobs that failed because of compile errors. [SCREENSHOT01]• Failed unit tests. [SCREENSHOT02]• Failure because of vulnerable packages. [SCREENSHOT03]• An alert from one of your failed builds. [SCREENSHOT04] <p>Evidence in your code that:</p> <ul style="list-style-type: none">• Compile errors have been fixed.• Unit tests have been fixed.• All critical security vulnerabilities caught by the "Analyze" job have been fixed.

CRITERIA	MEETS SPECIFICATIONS
<p>Utilize a configuration management tool to accomplish deployment to cloud-based servers.</p>	<p>Console output of appropriate failure for infrastructure creation job (using CloudFormation). [SCREENSHOT05]</p> <p>Console output of a smoke test job that is failing appropriately. [SCREENSHOT06]</p> <p>Console output of a successful rollback after a failed smoke test. [SCREENSHOT07]</p> <p>Console output of successful promotion of new version to production in CloudFront. [SCREENSHOT08]</p> <p>Console output of successful cleanup job that removes old S3 bucket and EC2 instance. [SCREENSHOT09]</p> <p>Evidence that deploy jobs only happen on <code>master</code> branch. [SCREENSHOT10]</p> <p>Evidence of deployed and functioning front-end application in an S3 bucket [URL02] and in CloudFront. [URL03]</p> <p>Evidence of healthy back-end application. [URL04]</p>

Section 3: Turn Errors into Sirens

CRITERIA	MEETS SPECIFICATIONS
Surface critical server errors for diagnosis using centralized logging.	<p>Evidence of Prometheus Server. [URL05]</p> <p>Evidence that Prometheus is monitoring memory, cpu and disk usage of EC2 instances. [SCREENSHOT11]</p> <p>Evidence that Prometheus and AlertManager send alerts when certain conditions exist in the EC2 instance. [SCREENSHOT12]</p>

Suggestions to Make Your Project Stand Out!

1. Add custom alert messages with more specific CI/CD pipeline failure reasons. The more specific the message, the faster the UdaPeople dev team can fix the issue.
 2. Use Circle CI locally to test your pipeline before pushing to git.
 3. Integrate other types of analysis or notification jobs.
 4. Write your own smoke tests to build more confidence before promoting the new build to production.
-