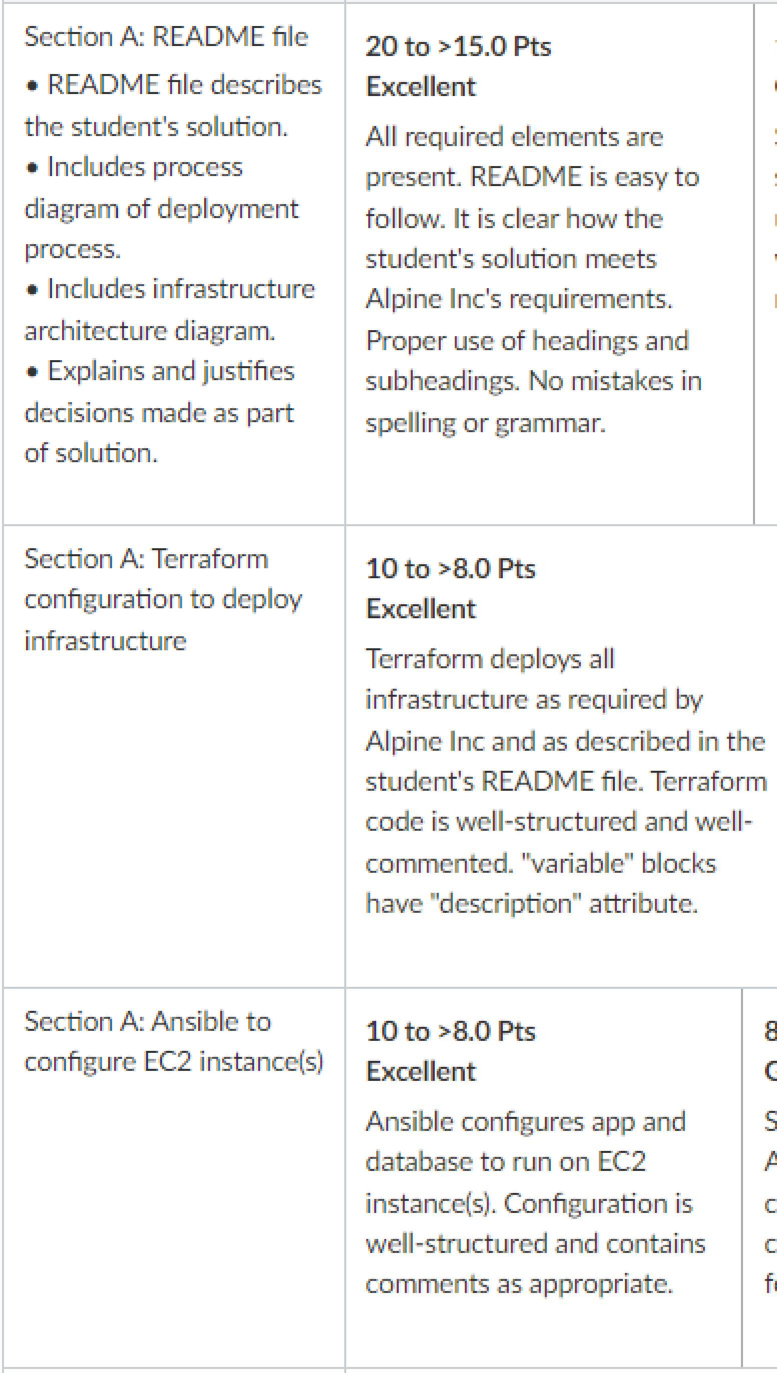
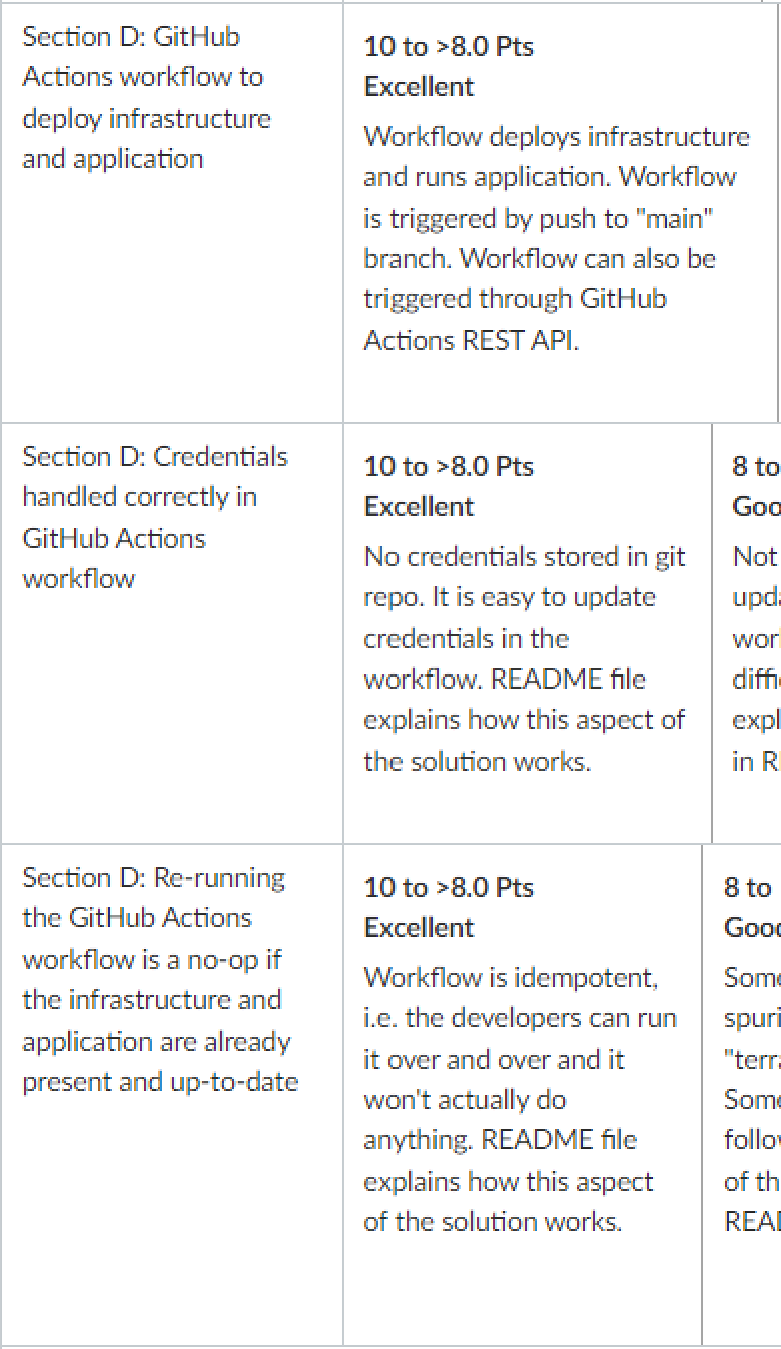


Section A: README file • README file describes the student's solution. • Includes process diagram of deployment process. • Includes infrastructure architecture diagram. • Explains and justifies decisions made as part of solution.	**20 to >15.0 Pts** **Excellent** All required elements are present. README is easy to follow. It is clear how the student's solution meets Alpine Inc's requirements. Proper use of headings and subheadings. No mistakes in spelling or grammar.
Section A: Terraform configuration to deploy infrastructure	**10 to >8.0 Pts** **Excellent** Terraform deploys all infrastructure as required by Alpine Inc and as described in the student's README file. Terraform code is well-structured and well-commented. "variable" blocks have "description" attribute.
Section A: Ansible to configure EC2 instance(s)	**10 to >8.0 Pts** **Excellent** Ansible configures app and database to run on EC2 instance(s). Configuration is well-structured and contains comments as appropriate.

Section A: Shell script to perform deployment of infrastructure and application Note: If you attempt the "HD" part of this assignment, also include a standalone shell script which performs the deployment. Use your README file to explain which file does what.	**10 to >8.0 Pts** **Excellent** Running a single shell script deploys the production infrastructure and runs the application on it. Shell script contains comments to explain what each part does and why. Uses environment variables for AWS credentials.	
Section B: Deploy app to 2x EC2 instances behind load balancer, with database on separate EC2 instance	**10 to >8.0 Pts** **Excellent** The Terraform configuration deploys 2x "app" EC2 instances behind a load balancer, plus a separate "database" EC2 instance. All EC2 instances are identical, except for whether they run the application or the database container. Ansible has separate roles for "app" and "database" hosts, with the correct container started up on each.	
Section C: Use remote backend for Terraform state Note: it is not necessary to use Terraform to deploy the state bucket itself.	**10 to >8.0 Pts** **Excellent** Terraform configuration contains the necessary code to use an S3 bucket as a remote state backend. The README file explains how this aspect of the solution works.	**8 to** **Goo** Som follo exp part file.



Section D: GitHub Actions workflow to deploy infrastructure and application	**10 to >8.0 Pts** **Excellent** Workflow deploys infrastructure and runs application. Workflow is triggered by push to "main" branch. Workflow can also be triggered through GitHub Actions REST API.	
Section D: Credentials handled correctly in GitHub Actions workflow	**10 to >8.0 Pts** **Excellent** No credentials stored in git repo. It is easy to update credentials in the workflow. README file explains how this aspect of the solution works.	**8 to** **Goo** Not upd wor diffi expl in R
Section D: Re-running the GitHub Actions workflow is a no-op if the infrastructure and application are already present and up-to-date	**10 to >8.0 Pts** **Excellent** Workflow is idempotent, i.e. the developers can run it over and over and it won't actually do anything. README file explains how this aspect of the solution works.	**8 to** **Good** Some spuri "terra Some follo of th REAI