**As per requirement for creation of scalable and secure cloud architecture for a** [**https://stylishop.com/ae/en**](https://stylishop.com/ae/en)**e-commerce platform.**

I would like to propose below solution.

In this solution we have consider application performance, security, scalability, CI/CD deployments.

With a web security scanner, Cloud Armor will protect DDOS attacks other attacks.

Front end web app and Backend web will configured behind VPC private network. The application will be open only via Cloud CDN and Loadbalancer configured behind cloud armor.

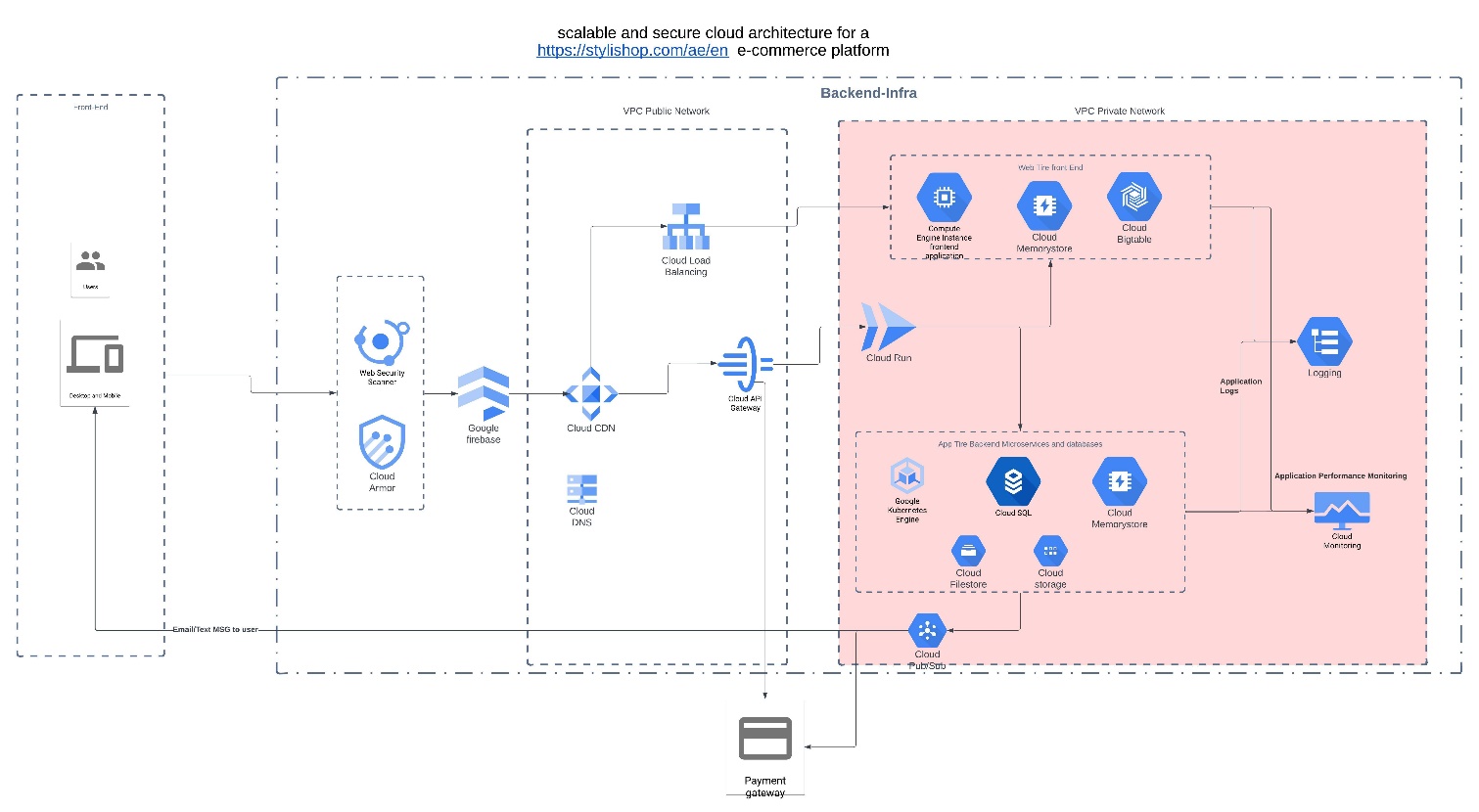
FrontEnd Web app is configured on Cloud compute instance with autoscaling and GKE is also configured with Karpenter for autoscaling.

For CI/CD we will use Argocd.

For Performance monitoring, profile management, of the application we will use Prometheus, Grafana, Loki and Pixie.

For Infrastructure performance monitoring will be monitor and alarms will be set in Google cloud logging.

1. **User Authentication:**
   * Users authenticate via Firebase Authentication.
   * User data will be stored in Cloud SQL
2. **WebApp access:**
   * Webapp can be access vis Cloud CDN
3. **Web Tier:**
   * The front end/UI is managed by Frontend webapp with Compute cloud instance with autoscaling.
   * Data is cached using Memory store (Redis).
   * Product catalog and other dynamic data are stored in Firestore or Bigtable.
4. **API Management:**
   * Payment process, order management, API requests are managed by API Gateway
5. **App Tier:**
   * Payment process, order management, user authentication, product catlos will be managed by multiple microservices.
   * Backend microservices are deployed on GKE or Cloud Run.
   * Functions of product catalog, and inventory user notification will be trigger and processed and other serverless executions are handled by Cloud Functions.
   * Data is cached using Memory store (Redis).
   * Structured data, such as orders and user profiles, are managed in Cloud SQL.
6. **File Storage:**
   * Static files, images, and backups are stored in Cloud Storage. File store will be used for temporary file processing.
7. **Messaging Queue:**
   * Pub/Sub is used for messaging and event-driven architecture.
8. **Search:**
   * Elasticsearch on GKE or Elastic Cloud for advanced search capabilities.
9. **Security**



Please find Terraform structure as below  
  
.

├── main.tf

├── variables.tf

├── outputs.tf

├── terraform.tfvars

├── modules

│ ├── vpc

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── compute\_instance

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── gke

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── cloud\_sql

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── memorystore

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── bigtable

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

│ ├── cloud\_functions

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── outputs.tf

└── └── cloud\_storage

│ ├── main.tf

│ ├── variables.tf

│ └── outputs.tf