We will divide the tasks into discrete portions, each of which will focus on a critical component of the suggested solution, in order to thoroughly handle the problem. An outline of the methodology and framework is provided here:   
  
  
**1. Evaluation of Costs:**

Costs of the Current Infrastructure:   
Determine the components of the infrastructure as it stands now (compute, storage, network, etc.).   
- Calculate these components' estimated expenses without mentioning any particular cloud providers.   
- Draw attention to the areas that have the most expense and room for savings.   
  
**2. Utilising Open Source Software**:

Suggested Open Source Software and Tools:   
- Compute Orchestration: Container orchestration using Kubernetes.   
- Configuration Management: Use Ansible to handle configurations.   
Jenkins is used for continuous integration and delivery (CI/CD).   
- Monitoring: Grafana and Prometheus for alerting and monitoring.   
- Database: Open-source database options include PostgreSQL and MySQL.   
- Terraform for infrastructure provisioning: Infrastructure as Code.   
 **3. Steps for Optimising Performance Plans:**

- Containerisation: Use Docker to containerise applications.  
- Caching: Use programmes like Varnish or Redis to implement caching.  
- Database Optimisation: Consistent query optimisation and database indexing.  
- Load Balancing: To achieve load balancing, use NGINX or HAProxy.  
- Resource Allocation: For dynamic resource allocation and scaling, use Kubernetes.  
  
**4. Strategy for Mitigating Downtime:**

Reducing Unavailable Time:  
- Blue-Green Deployment: To minimise downtime during updates, implement the blue-green deployment approach.  
- Rolling Updates: For rollback and rolling update techniques, use Kubernetes.  
- Automation: Use Jenkins pipelines to automate deployment and rollback procedures.

**5. The Strategy of Availability** :

Making Certain High Availability   
- Redundancy: For cluster failover and redundancy, use Kubernetes.   
- Replication: Use PostgreSQL replication or other database replication techniques.   
- Monitoring and Alerting: Configure Grafana and Prometheus to provide proactive alerting and monitoring.   
- Disaster Recovery: Make use of open-source technologies such as Restic to carry out backup and disaster recovery plans.   
  
**6. Timeline for Implementation:**

Plan for Phased Implementation:   
 Phase 1: Planning and Assessment   
Evaluate the infrastructure as it stands.   
- Arrange schedules for migration and setup.   
 Phase 2: Configuration and Setup of Tools   
- Configure Terraform, Ansible, and Jenkins.   
Set up a Kubernetes cluster.   
Phase 3: Testing and Migration   
- Convert apps to use containers.   
Examine deployment and monitoring configurations.   
Phase 4: Scaling and Optimisation   
- Enhance the performance of applications and databases.   
- Put scaling plans into action.   
Phase 5: Ongoing Enhancement   
- Track expenditures and performance.   
- Optimise continuously using metrics.

**7. Evaluation of Costs:**  
Comparing Projected and Current Costs:   
-Compare the operating costs of the current configuration with the new infrastructure that uses open-source tools.   
-Emphasise the possible cost savings from lower licencing costs, more effective use of resources, and increased productivity.