

Overview

This practical assessment evaluates your multi-cloud engineering expertise through hands-on implementation.

Deploy a real-world application across AWS, Azure, and Google Cloud Platform using only free tier services. Record your entire process and provide working infrastructure code.

Duration: 3 days

Format: Screen recording of live demo outlining architectural decisions and thought process

Application: Preferably a Laravel application, or alternatively a WordPress CMS — chosen for its practical use and compatibility with the free tier.

Cost: \$0 - All implementations must use verified free tier resources

Prerequisites

- AWS, Azure, and GCP accounts with active free tier eligibility
- Local development environment with required CLI tools
- Git repository for code submission
- Screen recording capability

Assessment Tasks

Task 1: Infrastructure as Code Multi-Cloud Deployment

Requirement: Deploy identical WordPress hosting infrastructure on AWS and Azure using Infrastructure as Code.

- There are two deployment requirements explained in the Task 2 and Task 3, implement the infrastructure as code required for Task 2 and Task 3.
- Pick the cloud providers optimum for the Task.

AWS Specifications:

- Use Terraform for all infrastructure provisioning
- Implement proper VPC networking with public/private subnets
- Configure Application Load Balancer with health checks
- Set up RDS MySQL with automated backups
- Implement proper security groups (least privilege)

Azure Specifications:

- Use Bicep or Terraform for infrastructure provisioning
- Use F1 App Service plan and B1ms MySQL database
- Implement Virtual Network with proper subnets
- Configure Basic Load Balancer or Application Gateway
- Set up Azure Database for MySQL with firewall rules
- Implement Network Security Groups

GCP Specifications:

- Use Terraform or Deployment Manager for all infrastructure provisioning
- Implement Virtual Network with proper subnets
- Use Compute Engine f1-micro and Cloud SQL db-f1-micro for free tier usage
- Provide Deployment Manager or Terraform configuration
- Implement proper security groups (least privilege)

Evaluation Criteria:

- Infrastructure code quality and organization
- Proper resource tagging and naming conventions
- Security configuration (IAM, network isolation, encryption)
- Cost optimization for free tier usage
- Documentation clarity and architectural decisions

Deliverables:

- Complete, executable Terraform code for the cloud provider or providers
- Architecture documentation with configuration examples
- README with deployment instructions and design rationale

Task 2: Kubernetes Container Orchestration

Requirement: Deploy WordPress as a containerized application using Kubernetes. Use AWS EKS or Azure AKS or GCP GKE and provide equivalent theoretical configuration implementations for the other cloud providers which you have not picked.

Technical Requirements:

- WordPress container with proper resource limits
- MySQL container with persistent storage
- Secrets management for database credentials
- Health checks and readiness probes
- Network policies for service communication

Evaluation Criteria:

- Kubernetes manifest/Helm quality and best practices
- Container security implementation - optional
- Resource management and optimization
- Understanding of cloud-specific Kubernetes features
- Troubleshooting and debugging capabilities

Deliverables:

- Working deployment with accessible WordPress site
- Complete Kubernetes manifests
- Custom Dockerfile if modifications made
- kubectl/helm commands used and troubleshooting notes

Task 3: Platform Services and Auto-scaling

Requirement: Deploy WordPress using each cloud provider's Platform-as-a-Service offerings. Implement auto-scaling and monitoring.

AWS Elastic Beanstalk:

- Deploy PHP application using Beanstalk
- Configure t2.micro instances with auto-scaling (max 2 instances)
- Implement CloudWatch monitoring and alarms - optional
- Set up a simple deployment pipeline

Azure App Service:

- Deploy to F1 Free tier App Service
- Configure deployment slots for staging
- Set up Application Insights monitoring - optional
- Implement auto-scaling rules
- Set up a simple deployment pipeline

GCP Cloud Run:

- Deploy containerized WordPress to Cloud Run
- Configure auto-scaling (0-10 instances)
- Set up Cloud Monitoring and alerting - optional
- Set up a simple deployment pipeline

Advanced Requirements (optional):

- Monitoring dashboard creation
- Log aggregation and analysis
- Performance optimization recommendations

Evaluation Criteria:

- Platform service configuration expertise
- Auto-scaling implementation and testing
- Monitoring and observability setup - optional
- Performance optimization techniques
- Understanding of platform limitations and workarounds

Deliverables:

- Working application demo on one of the cloud providers.
- Auto-scaling configuration and test results
- Monitoring dashboards and alert configurations (optional)
- Optimization recommendations for a production deployment

Task 4: Security, Monitoring, and Cost Optimization - Optional

Requirement: Implement security measures, monitoring solutions, and cost optimization strategies across all deployments.

Security Implementation:

- IAM roles and policies with least privilege access
- Network security groups and firewall rules
- SSL/TLS certificate configuration
- Secrets management implementation
- Security scanning and vulnerability assessment

Monitoring Setup:

- Centralized logging across all platforms
- Application performance monitoring
- Infrastructure monitoring and alerting
- Cost monitoring and budget alerts
- SLA/SLO definition and tracking

Cost Optimization:

- Resource utilization analysis
- Right-sizing recommendations
- Automation for resource cleanup
- Cost comparison across platforms
- Free tier usage optimization

Evaluation Criteria:

- Security best practices implementation
- Comprehensive monitoring strategy
- Cost awareness and optimization
- Automation and operational excellence
- Incident response preparedness

Deliverables:

- Security configuration documentation
- Monitoring dashboards and alert policies
- Cost analysis and optimization report
- Operational runbooks and procedures

Technical Specifications

Only use Free Tier Resources

AWS (12 months):

- EC2: 750 hours t2.micro instances
- RDS: 750 hours db.t3.micro MySQL instances
- ELB: 750 hours Application Load Balancer
- EBS: 30GB General Purpose SSD storage

Azure (12 months + always free):

- App Service: F1 tier (always free)
- Virtual Machines: 750 hours B1s instances
- Database: B1ms MySQL (12 months free)
- Load Balancer: Basic tier (always free)

GCP (Always free + \$300 credit):

- Compute Engine: 1 f1-micro instance (always free)
- Cloud SQL: db-f1-micro MySQL (always free)
- Cloud Run: 2M requests/month (always free)
- Cloud Storage: 5GB regional storage (always free)

Instance Types

Use only instance types to avoid charges:

- AWS: t2.micro (EC2), db.t3.micro (RDS)
- Azure: F1 (App Service), B1s (VM), B1ms (Database)
- GCP: f1-micro (Compute), db-f1-micro (Cloud SQL)

Evaluation Criteria

Technical Excellence (40%)

- Infrastructure code quality and best practices
- Security implementation and configuration
- Performance optimization and monitoring
- Error handling and resilience

Multi-Cloud Expertise (30%)

- Understanding of service equivalencies across platforms
- Platform-specific optimization techniques
- Migration strategies and considerations
- Cost-aware architectural decisions

Problem-Solving Skills (20%)

- Creative solutions within free tier constraints
- Troubleshooting methodology and documentation
- Performance tuning and optimization
- Scalability considerations

Communication and Documentation (10%)

- Clear explanations of technical decisions
- Quality of code comments and documentation
- Presentation skills in recorded demo
- Time management and organization

Submission Requirements

Well structured repository

Documentation Requirements

- Architecture diagrams for each platform (maximum of two diagrams for the cloud providers you have picked is sufficient)
- Security assessment and compliance checklist
- Production deployment cost optimization recommendations, if any.
- Operational procedures and readme/runbooks

Video Recording Requirements

- Minimum 720p resolution with clear audio
- Demo with verbal explanation
- Problem-solving process demonstration
- Final working demo of all implementations

Success Criteria

Passing Requirements:

- Infrastructure code must be executable and well-documented
- Security configurations must follow industry best practices
- Implementations must stay within free tier limits
- Recording must demonstrate implementation process

Excellence Indicators:

- Innovative solutions to free tier limitations
- Advanced security configurations
- Performance optimization techniques
- Clear understanding of multi-cloud trade-offs

Focus on demonstrating deep technical knowledge, problem-solving skills, and practical implementation expertise.