DevOps Shack

50 DevOps Tasks For Interview Preparation

1. **Continuous Integration Pipeline**: Set up a CI pipeline using tools like Jenkins or GitLab CI to automate the build, test, and deployment process of a sample application.

2. **Infrastructure as Code (IaC)**: Develop infrastructure using Terraform or CloudFormation to provision resources on AWS, Azure, or Google Cloud.

3. **Container Orchestration**: Build a Kubernetes cluster and deploy microservices-based applications on it.

4. **Configuration Management**: Implement configuration management using Ansible or Puppet to maintain server configurations consistently.

5. **Monitoring and Logging**: Set up monitoring and logging using tools like Prometheus, Grafana, ELK stack (Elasticsearch, Logstash, Kibana), or Splunk for real-time analysis.

6. **Automated Testing**: Develop automated testing frameworks using tools like Selenium for web applications or JUnit for Java applications.

7. **Deployment Automation**: Create deployment pipelines using tools like Spinnaker or Harness to automate application deployments to various environments.

8. **Infrastructure Monitoring**: Build a monitoring system using Nagios or Zabbix to monitor server health and performance metrics.

9. **Containerization**: Containerize applications using Docker and deploy them to a container orchestration platform like Kubernetes or Docker Swarm.

10. **Blue/Green Deployment**: Implement blue/green deployment strategies using tools like AWS CodeDeploy or Kubernetes to minimize downtime during deployments.

11. **Serverless Architecture**: Develop serverless applications using AWS Lambda, Azure Functions, or Google Cloud Functions and deploy them using CI/CD pipelines.

12. **Security Automation**: Implement security automation using tools like SonarQube or OWASP ZAP to identify and remediate security vulnerabilities in code.

13. **Continuous Delivery Pipeline**: Create a complete CD pipeline from code commit to production deployment using Jenkins, GitLab CI/CD, or Azure DevOps.

14. **Immutable Infrastructure**: Build immutable infrastructure using Packer to create machine images and deploy them using Terraform or AWS CloudFormation.

15. **Microservices Architecture**: Design and implement a microservices-based architecture using Spring Boot, Docker, and Kubernetes.

16. **Disaster Recovery**: Develop disaster recovery plans and automate recovery processes using tools like AWS Backup or Azure Site Recovery.

17. **GitOps**: Implement GitOps practices using tools like Flux or Argo CD to manage Kubernetes deployments through Git repositories.

18. **Infrastructure Monitoring**: Set up infrastructure monitoring using Prometheus, Grafana, and Alertmanager for alerting and visualization.

19. **High Availability**: Design and implement highly available architectures using load balancers, auto-scaling groups, and multi-region deployments.

20. **Cloud Cost Optimization**: Implement cost optimization strategies using AWS Cost Explorer or Azure Cost Management to monitor and optimize cloud spending.

21. **Service Mesh**: Implement a service mesh using Istio or Linkerd to manage microservices communication, security, and observability.

22. **CI/CD for Mobile Apps**: Create CI/CD pipelines for mobile apps using tools like Fastlane for iOS or Bitrise for Android.

23. **Infrastructure Testing**: Perform infrastructure testing using tools like Serverspec or InSpec to ensure that infrastructure configurations meet compliance requirements.

24. **Secrets Management**: Implement secrets management using tools like HashiCorp Vault or AWS Secrets Manager to securely store and manage sensitive information.

25. **ChatOps**: Integrate chat platforms like Slack or Microsoft Teams with deployment pipelines for real-time notifications and interaction.

26. **Continuous Compliance**: Implement continuous compliance checks using tools like Chef Compliance or AWS Config to ensure that infrastructure configurations adhere to security policies.

27. **Serverless CI/CD**: Implement CI/CD pipelines for serverless applications using AWS CodePipeline or Azure Pipelines.

28. **Self-Healing Infrastructure**: Implement self-healing mechanisms using tools like Kubernetes Horizontal Pod Autoscaler or AWS Auto Scaling to automatically adjust resources based on demand.

29. **Configuration Drift Detection**: Set up configuration drift detection using tools like AWS Config or Ansible Tower to identify inconsistencies in infrastructure configurations.

30. **Immutable Deployments**: Implement immutable deployments using tools like AWS CodeDeploy or Kubernetes to deploy new versions of applications without modifying existing instances.

31. **Compliance as Code**: Define compliance rules as code using tools like Terraform Compliance or AWS Config Rules to enforce regulatory requirements.

32. **Continuous Security**: Integrate security scanning tools like OWASP ZAP or Clair into CI/CD pipelines to detect vulnerabilities in code and dependencies.

33. **Automated Rollback**: Implement automated rollback mechanisms using Jenkins or GitLab CI/CD to revert deployments in case of failures.

34. **Multi-Cloud Deployments**: Design and implement deployments across multiple cloud providers using tools like Terraform or Kubernetes.

35. **Serverless Data Processing**: Develop serverless data processing pipelines using AWS Glue or Google Cloud Dataflow for ETL (Extract, Transform, Load) tasks.

36. **Immutable Infrastructure Testing**: Perform testing on immutable infrastructure using tools like Serverspec or InSpec to ensure that changes don't introduce configuration drift.

37. **Configuration Validation**: Implement configuration validation checks using tools like Chef InSpec or Terraform to verify that infrastructure configurations meet compliance requirements.

38. **Secrets Rotation**: Automate secrets rotation using tools like AWS Secrets Manager or HashiCorp Vault to regularly update credentials and keys.

39. **Canary Deployment**: Implement canary deployment strategies using tools like Istio or AWS App Mesh to gradually roll out new versions of applications and monitor for issues.

40. **Self-Service Infrastructure**: Build self-service infrastructure provisioning portals using tools like Terraform Enterprise or AWS Service Catalog for streamlined resource allocation.

41. **Serverless API**: Develop serverless APIs using AWS API Gateway, Azure API Management, or Google Cloud Endpoints and deploy them using CI/CD pipelines.

42. **Continuous Compliance Remediation**: Integrate remediation actions into CI/CD pipelines using tools like Ansible or AWS Systems Manager to automatically address compliance violations.

43. **Event-Driven Architecture**: Design and implement event-driven architectures using message brokers like Apache Kafka or AWS SNS/SQS.

44. **Observability as Code**: Define observability configurations as code using tools like Prometheus Operator or AWS CloudFormation to automate monitoring setup.

45. **Infrastructure Scalability Testing**: Perform scalability testing using tools like Locust or Apache JMeter to ensure that infrastructure can handle expected loads.

46. **Infrastructure Dependency Mapping**: Map dependencies between infrastructure components using tools like AWS X-Ray or Google Cloud Trace for improved troubleshooting.

47. **Serverless Cron Jobs**: Implement serverless cron jobs using AWS Lambda or Azure Functions for scheduled tasks and automation.

48. **Multi-Cloud Disaster Recovery**: Design and implement disaster recovery plans across multiple cloud providers for increased resilience.

49. **Federated Identity Management**: Implement federated identity management using tools like AWS Cognito or Azure Active Directory for centralized authentication and authorization.

50. **Chaos Engineering**: Conduct chaos engineering experiments using tools like Chaos Monkey or Gremlin to proactively identify weaknesses in infrastructure resilience.