DevOps Interview Questions and Answers

1. What are the common day-to-day activities in DevOps when working in the cloud?

Daily activities often include managing CI/CD pipelines, monitoring system performance, infrastructure provisioning using Infrastructure as Code (IaC) tools like Terraform, collaborating with developers to troubleshoot build and deployment issues, implementing security protocols, managing container orchestration with Kubernetes and Docker, and handling incident response and root cause analysis.

2. How do you effectively use Kubernetes and Docker in your daily DevOps tasks?

Docker is used to containerize applications, ensuring they run consistently across environments. Kubernetes orchestrates these containers, managing deployments, scaling, and ensuring high availability. I use Kubernetes to manage rolling updates, maintain pod health, and automate scaling. Additionally, ConfigMaps, Secrets, and networking policies are used for better configuration and security management.

3. What happens internally when a container runs in Kubernetes?

When a container runs in Kubernetes:

- The `kubectl` command interacts with the API server.
- The scheduler decides the best node for the pod based on resources and constraints.
- The kubelet on the chosen node communicates with the container runtime (e.g., Docker) to pull the image and start the container.
- Networking is configured using the CNI plugin, and services connect pods to each other and external traffic.
- The controller manager maintains the desired state by monitoring the pod?s status and restarting it if needed.

4. What are the key security features available in Kubernetes?

Kubernetes provides several security features, including:

- Role-Based Access Control (RBAC) for managing permissions.
- Network Policies to control traffic between pods.
- Secrets Management for securely storing sensitive data.
- Pod Security Policies (PSPs) to enforce security rules on pods.
- Service Accounts for granting permissions to pods.

- TLS/SSL encryption for securing communication between components.
- API auditing for tracking and reviewing access.

5. Can you explain the main components of Kubernetes?

Kubernetes is composed of:
- Master Node Components:
  - API Server: Central management point for the cluster.
  - Scheduler: Assigns pods to available nodes.
  - Controller Manager: Maintains the desired state of resources.
  - etcd: Key-value store for cluster configuration data.
- Worker Node Components:
  - Kubelet: Manages pod lifecycle.
  - Kube-proxy: Manages network rules and load balancing.
  - Container Runtime: Runs the containers.

6. What are exit codes and their importance?

Exit codes are numerical codes returned by a process indicating its termination status. Common exit codes include:
- 0: Successful execution.
- 1: General error.
- 137: Process killed (often due to memory constraints).
- 139: Segmentation fault.

7. How would you troubleshoot an unscheduled pod in Kubernetes?

Steps include:
- Describe the pod using `kubectl describe pod <pod-name>` for error messages.
- Check node status with `kubectl get nodes` and `kubectl describe node <node-name>`.
- Verify resource constraints, affinity rules, and taints/tolerations.
- Review events and check for scheduling issues (e.g., insufficient resources).

8. Can you describe your experience with incident management and how you categorize incident priority?

Yes, I have experience handling incidents using ITIL frameworks. Prioritization is based on:
- P1 (Critical): Major disruption with no workaround.
- P2 (High): Significant impact with a potential workaround.

- P3 (Medium): Limited impact, minor inconvenience.
- P4 (Low): Cosmetic or minor issue.

9. How do Prometheus and Grafana interact, and what is Prometheus' data source?

Prometheus scrapes metrics from monitored targets using exporters. Grafana connects to Prometheus as a data source, querying and visualizing the collected metrics to create dashboards.

10. Can you explain Linux mechanisms during system startup?

The startup process includes:
- Bootloader (e.g., GRUB) loading the kernel.
- Kernel initialization, setting up hardware and mounting the root filesystem.
- init/Systemd starting system services based on configuration files.

11. How do you enable internal communication between multiple AWS accounts?

Options include:
- AWS Transit Gateway for scalable, hub-and-spoke VPC connectivity.
- VPC Peering for direct VPC communication.
- Shared VPCs and AWS Organizations for centralized resource sharing.

12. What is the difference between IR (Incident Report) and SR (Service Request)?

Incident Report (IR): Documentation of unexpected disruptions.
Service Request (SR): Routine user requests like password resets or new access.

13. Do you have experience with monitoring? What tools have you used?

Yes, I have extensive experience with tools like Prometheus, Grafana, ELK Stack, and AWS CloudWatch for real-time monitoring, visualization, and alerting.

14. What are your main activities related to monitoring?

Setting up monitoring dashboards, defining alerts, analyzing logs, creating custom metrics, and ensuring systems meet performance and availability SLAs.

15. What is the difference between logs and metrics?

Logs: Detailed, time-stamped event records from systems and applications.
Metrics: Aggregated, quantitative measurements like CPU usage or request latency, usually tracked over time.

16. What would you do if you noticed high utilization in an application on your monitoring dashboard?

The first step is to identify the resource causing the spike (CPU, memory). Depending on findings, I would:

- Scale resources or pods.
- Optimize the application or load balancing.
- Redistribute workloads across nodes.

17. Do you have experience creating monitors?

Yes, I have created custom monitors and alerts in Prometheus, CloudWatch, and other tools for tracking key metrics and ensuring system reliability.

18. What is a sidecar container in Kubernetes, and what are its use cases?

A sidecar container runs alongside the main application container within the same pod, providing supplementary functionality like logging, monitoring, or proxying. Use cases include service meshes, log aggregators, and backup agents.

19. Do you have experience with Infrastructure as Code (IaC) tools like Terraform?

Yes, I have experience using Terraform for defining and provisioning cloud infrastructure through declarative code, enabling version-controlled, automated deployments.

20. Can you provide a sample Dockerfile to create an Nginx image?

```
FROM nginx:latest
COPY ./index.html /usr/share/nginx/html/index.html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

21. What do you know about testing in DevOps?

Testing in DevOps includes unit, integration, and end-to-end testing integrated into CI/CD pipelines. Tools like JUnit, Selenium, and JMeter ensure code quality, performance, and functionality are maintained during the deployment process.