Name – Bhavani Prasad Moru.

Dockerfile for nginx 1.19

Write a Dockerfile to run nginx version 1.19 in a container. Choose a base image, considering security best practices, and aim for the image to pass a container image security test  
Your answer

**To create a secure Docker base image, start with a minimal distribution like Alpine or distroless images to reduce the attack surface. Implement multi-stage builds to separate build dependencies from the runtime environment, ensuring only necessary components are included in the final image. Run containers as non-root users with the least privileges required by creating dedicated service accounts and setting appropriate file permissions. Remove unnecessary tools, shell access, and debugging utilities from production images. Keep the image updated with security patches by regularly rebuilding with the latest base images. Use specific version tags rather than 'latest' to ensure predictability. Implement proper secrets management by avoiding embedding sensitive data in the image. Configure proper network security by exposing only necessary ports and implementing reasonable timeouts. Finally, scan images with security tools like Trivy, before deployment to identify and remediate vulnerabilities.**

Kubernetes StatefulSet

Write a Kubernetes StatefulSet to deploy the nginx container from the previous question. Utilize persistent volume claims and define resource limits for optimal performance.  
Your answer

**The Kubernetes StatefulSet configuration deploys a secured NGINX 1.19 container across multiple replicas with consistent networking identities. It implements resource management with appropriate CPU and memory limits (500m CPU/256Mi memory) and requests (100m CPU/128Mi memory) to ensure stable performance. Each pod uses persistent storage through volumeClaimTemplates that provision 1Gi of storage for website content. The configuration implements security best practices by running containers as a non-root user and including health checks through readiness and liveness probes. Instead of embedding the NGINX configuration directly in the manifest, the solution leverages an external configuration file (nginx\_app.conf) that's loaded into Kubernetes as a ConfigMap. To create this ConfigMap, you run the command kubectl create configmap nginx-config --from-file=default.conf=./nginx\_app.conf, which imports the local nginx\_app.conf file into Kubernetes and makes it available for mounting into the pods. This approach provides cleaner separation of concerns, allowing the NGINX configuration to be managed independently from the deployment configuration.**

Build a Deployment Pipeline

Set up a streamlined build and deployment pipeline for the nginx application using GitHub Actions or an equivalent CI/CD tool. Ensure the pipeline covers building the Docker image, running security checks, and deploying to a Kubernetes cluster.

Your answer

**The CI/CD process in this GitHub Actions workflow creates a comprehensive and secure pipeline for your Nginx application. When code is pushed to any branch, the workflow starts by performing static code analysis with SonarQube and vulnerability scanning with OWASP Dependency Check to ensure code quality and security. For release branches specifically, the pipeline extracts the application version from package.json and combines it with a shortened commit hash to create semantically versioned Docker image tags.**

**The build process uses JFrog Artifactory as the container registry and pulls base images from the same internal repository for consistency and security. All sensitive information, including API keys and credentials, is securely retrieved from HashiCorp Vault rather than being stored as GitHub secrets. Once built, the Docker image undergoes security scanning with Trivy to detect vulnerabilities before publishing to the registry.**

**For release branches, the pipeline creates a detailed artifact containing image information including the full image name, tag, commit hash, branch details, and build timestamp. This information is crucial for tracking which version of the application is in which environment. The pipeline intentionally omits deployment tasks, focusing solely on producing secure, properly tagged Docker images. This separation of concerns allows deployment to be managed independently using the published image information, enabling different teams to handle deployments when ready without coupling it to the build process.**

**The CI/CD implementation comprises two distinct pipelines that work seamlessly together in a GitOps workflow. The CI pipeline focuses on code quality and secure container creation, running SonarQube and OWASP scans before building Docker images with semantic versioning tags (combining app version and commit hash) and securing them through Trivy vulnerability scanning. All credentials and sensitive data are stored in HashiCorp Vault rather than as pipeline secrets. The CD pipeline, built on ArgoCD, takes a declarative approach where infrastructure changes flow through Git commits rather than direct cluster manipulation. When deploying, the pipeline updates image references in a GitOps repository, which ArgoCD automatically syncs to the appropriate Kubernetes environment (dev, staging, or production). This separation of build and deployment concerns enables progressive delivery with proper validation at each stage, comprehensive audit trails through Git history, and a self-healing infrastructure that automatically reconciles to the desired state.**

Text Manipulation Problem

Choose or create a text manipulation problem that involves using awk, sed, tr, and/or grep. Solve the problem, considering efficiency and readability.

Your answer

**This script extracts ERROR and WARN logs from application logs, formats them as CSV with timestamp, level, service, email, and IP columns, and sorts by timestamp. It uses grep to filter by log level, sed to extract fields using regex, awk to clean service names and handle missing IPs, and sort to arrange entries chronologically.**

Text Manipulation with an Object Orientated Programming Language

Solve the text manipulation problem from the previous question using any Object Orientated Programming language of your choice. Provide a clear and well-documented solution.

Your answer

**This Python solution uses an object-oriented approach to process application logs, extracting ERROR and WARN entries. The code defines two main classes: LogEntry for representing individual log records with parsed fields, and LogProcessor which handles file reading, regex parsing, and CSV output. The solution uses encapsulation to organize related data and behaviors together, implements sorting through custom comparison methods, and provides proper error handling. Unlike the shell script version, this implementation offers better maintainability, extensibility, and clearer structure, making it easier to add features like additional filtering options or alternative output formats.**

Sum of Even Fibonacci Numbers

Write a program in a Object Orientated Programming language of your choice to calculate the sum of the first 100 even-valued Fibonacci numbers. Consider efficiency and demonstrate good coding practices.

Your answer

**I implemented a solution for calculating the sum of the first 100 even-valued Fibonacci numbers using object-oriented programming. My approach uses the mathematical property that every third Fibonacci number is even, which allows me to efficiently generate only the even numbers in the sequence. By using the formula F(n+3) = 4\*F(n) + F(n-3), the algorithm directly computes each even Fibonacci number without calculating the odd-valued numbers in between, significantly reducing the computational work. The EvenFibonacciCalculator class encapsulates all the necessary state variables and methods, with proper initialization and a clean interface for generating numbers and calculating sums. This implementation achieves O(n) time complexity with O(1) space complexity, making it highly efficient even for large values of n.**

Intersection of Sorted Arrays

Write a function in a Object Orientated Programming language of your choice that takes two sorted arrays of integers as input and returns an array containing numbers common to both arrays without duplicates.

Your answer

**I implemented a solution for finding the intersection of two sorted arrays using object-oriented programming. My approach uses a two-pointer technique to efficiently traverse both arrays simultaneously, which achieves optimal O(n+m) time complexity where n and m are the lengths of the arrays.**

**The solution includes an ArrayIntersection class with a findIntersection method that handles the core logic. The algorithm works by maintaining two pointers, one for each array, and advancing them strategically based on comparing elements. When matching elements are found, they're added to the result set. A key optimization is skipping duplicate values in the input arrays to ensure the output contains only unique elements.**

**I also included an alternative implementation using a HashSet approach, which can be more versatile for unsorted arrays but requires additional space. This demonstrates flexibility in solving the problem while maintaining the object-oriented structure.**

**The implementation handles edge cases gracefully, including empty arrays or arrays with no intersection. For demonstration purposes, I included a main method with test cases that show the solution working correctly with different input scenarios. The code is thoroughly documented with comments explaining the approach and logic behind each step.**

Decimal Digit Transformation

Write a function in an Object Orientated Programming language of your choice that, when passed a decimal digit X, calculates and returns the value of X + XX + XXX + XXXX. For example, if X is 3, the function should return 3702 (3 + 33 + 333 + 3333). Ensure the function handles valid inputs and provides meaningful error messages for invalid inputs.

Your answer

**I implemented a solution for the decimal digit transformation problem using object-oriented programming. My DigitTransformer class calculates the sum of 3+33 + 333 +3333 for a given decimal digit.**

**For example, when X=3:**

* **First term: 3**
* **Second term: 33**
* **Third term: 333**
* **Fourth term: 3333**
* **Sum: 3 + 33 + 333 + 3333 = 3702**

**The implementation validates that inputs are single decimal digits (0-9) and throws appropriate errors for invalid values. I also included an optimized version that uses the mathematical insight that this pattern equals X\*1234, which is more efficient while producing identical results.**