Azure DevOps Expert

AZ-400

Exam Questions

Nensi Ravaliya

→

Microsoft Azure DevOps Engineer Expert (AZ-400) Exam Questions

200 Practice Questions

Domain 1: Design and Implement Processes and Communications (10-15%)

Question 1

Q: What is the primary purpose of Azure Boards in Azure DevOps?

A: Azure Boards provides work item tracking and agile planning tools, including Kanban boards, backlogs, sprint planning, and customizable dashboards for managing project tasks, bugs, and user stories.

Question 2

Q: Which Azure DevOps feature allows you to link work items to code commits and pull requests?

A: Traceability features in Azure Repos allow linking work items to commits, pull requests, and builds using work item IDs prefixed with # (e.g., #123).

Question 3

Q: What is the recommended branching strategy for large teams using Git in Azure DevOps?

A: Git Flow or GitHub Flow are recommended. Git Flow uses feature, develop, release, and hotfix branches. GitHub Flow is simpler with feature branches merged directly to main.

Question 4

Q: How can you configure notifications for pipeline events in Azure DevOps?

A: Navigate to Project Settings > Notifications to configure email alerts for build completions, failures, and other pipeline events. Team and personal subscriptions can be customized.

Question 5

Q: What is the purpose of Azure DevOps dashboards?

A: Dashboards provide customizable views of project metrics, build status, work item progress, test results, and other KPIs using configurable widgets.

Question 6

Q: Which chart widget should you use to track team velocity in Azure Boards?

A: The Velocity widget shows the team's velocity over sprints, displaying completed story points or work items per iteration.

Question 7

Q: How do you implement feedback cycles in Azure DevOps?

A: Use Azure Test Plans for user acceptance testing, GitHub issues or Azure Boards for feedback collection, and Application Insights for production feedback through telemetry.

Question 8

Q: What is the purpose of process templates in Azure DevOps?

A: Process templates (Agile, Scrum, CMMI, Basic) define work item types, workflows, and fields available in a project, enabling customization of how teams track work.

Question 9

Q: How can you integrate Azure Boards with GitHub?

A: Install the Azure Boards app from GitHub Marketplace, then link repositories to Azure Boards projects for bidirectional work item and commit linking.

Question 10

Q: What is a burn-down chart used for in Azure DevOps?

A: A burn-down chart shows remaining work over time during a sprint, helping teams track progress toward completing sprint goals and identify potential delays.

Domain 2: Design and Implement a Source Control Strategy (10-15%)

Question 11

Q: What Git command initializes a new local repository?

A: `git init` creates a new Git repository in the current directory, initializing the .git folder for version control.

Question 12

Q: How do you clone an Azure Repos Git repository?

A: Use `git clone <repository-url>` where the URL is obtained from Azure Repos > Clone button. Authentication can use PAT, SSH keys, or credential manager.

Question 13

Q: What is the purpose of branch policies in Azure Repos?

A: Branch policies enforce code quality by requiring pull request reviews, build validation, linked work items, comment resolution, and merge strategies before completing PRs.

Question 14

Q: How do you configure a minimum number of reviewers for pull requests?

A: In Branch Policies, enable "Require a minimum number of reviewers" and specify the count. You can also require approval from specific users or groups.

Question 15

Q: What is Git LFS and when should you use it?

A: Git Large File Storage (LFS) replaces large files with text pointers while storing file contents on a remote server. Use for binary files, media, or files exceeding 100MB.

Question 16

Q: How do you recover a deleted branch in Azure Repos?

A: Navigate to Repos > Branches, click “Search” with the deleted branch name, then click “Restore” from the context menu to recover recently deleted branches.

Question 17

Q: What is the difference between git merge and git rebase?

A: `git merge` creates a merge commit preserving branch history. `git rebase` rewrites commit history by replaying commits on top of another branch for linear history.

Question 18

Q: How do you squash commits when completing a pull request?

A: In Branch Policies, enable “Limit merge types” and select “Squash merge” to combine all PR commits into a single commit when merging.

Question 19

Q: What command purges sensitive data from Git history?

A: Use `git filter-branch` or `git-filter-repo` (recommended) to rewrite history and remove sensitive files. BFG Repo-Cleaner is another option for large repositories.

Question 20

Q: How do you configure Git credential helpers?

A: Use `git config --global credential.helper <helper-name>` where helper can be wincred (Windows), osxkeychain (Mac), or cache/store (Linux).

Question 21

Q: What is a Git submodule?

A: A submodule embeds an external repository within another repository at a specific commit, allowing you to include dependencies while maintaining separate version control.

Question 22

Q: How do you enable Git hooks in Azure Repos?

A: Server-side hooks are configured through Branch Policies. Client-side hooks are local scripts in .git/hooks folder that run on events like pre-commit or pre-push.

Question 23

Q: What is the purpose of .gitignore files?

A: .gitignore specifies files and patterns that Git should ignore and not track, typically used for build outputs, dependencies, secrets, and IDE-specific files.

Question 24

Q: How do you configure required status checks for branches?

A: In Branch Policies, enable "Build validation" and select required builds that must succeed before PR completion. Additional status checks from external services can be required.

Question 25

Q: What is Git Scalar and when should you use it?

A: Git Scalar is a tool for managing large Git repositories, enabling features like sparse-checkout, partial clone, and file system monitor for improved performance.

Domain 3: Design and Implement Build and Release Pipelines (50-55%)

Question 26

Q: What is the difference between Classic and YAML pipelines in Azure DevOps?

A: Classic pipelines use a visual designer stored in Azure DevOps. YAML pipelines are code-based, stored in the repository, enabling version control and code review for pipelines.

Question 27

Q: How do you trigger a pipeline on push to specific branches?

A: In YAML, use the trigger section: `trigger: branches: include: - main - develop` or exclude branches with the exclude keyword.

Question 28

Q: What is a multi-stage YAML pipeline?

A: A multi-stage pipeline defines multiple stages (build, test, deploy) in a single YAML file with dependencies, conditions, and approvals between stages.

Question 29

Q: How do you create reusable pipeline templates?

A: Create template YAML files with parameters and reference them using `template: template-file.yml@repository` with parameter values passed during reference.

Question 30

Q: What is the purpose of Azure Artifacts?

A: Azure Artifacts hosts package feeds for NuGet, npm, Maven, Python, and Universal packages, enabling teams to share and version dependencies across projects.

Question 31

Q: How do you configure pipeline caching?

A: Use the Cache task with a key based on files (like package-lock.json) and a path to cache. Caching restores files between pipeline runs to speed up builds.

Question 32

Q: What is a deployment group in Azure DevOps?

A: A deployment group is a collection of target machines with agents installed, used for deploying to multiple on-premises servers or VMs in release pipelines.

Question 33

Q: How do you implement blue-green deployment using Azure App Service?

A: Create a staging deployment slot, deploy to staging, verify, then use the App Service Manage task to swap slots, making staging the production environment.

Question 34

Q: What is a canary deployment strategy?

A: Canary deployment gradually releases changes to a small subset of users first, monitoring for issues before rolling out to the entire user base.

Question 35

Q: How do you configure approval gates for deployments?

A: In release pipelines or YAML environments, add pre-deployment approvals specifying approvers, timeout, and instructions. Deployment pauses until approved.

Question 36

Q: What is the purpose of release gates?

A: Release gates are automated checks (Azure Monitor alerts, work item queries, REST APIs) that must pass before deployment proceeds, ensuring quality and compliance.

Question 37

Q: How do you reference secrets from Azure Key Vault in pipelines?

A: Create a variable group linked to Key Vault, or use the Azure Key Vault task to download secrets as pipeline variables at runtime.

Question 38

Q: What is the difference between Microsoft-hosted and self-hosted agents?

A: Microsoft-hosted agents are managed VMs provided by Microsoft with pre-installed tools. Self-hosted agents are your own machines with custom software and configurations.

Question 39

Q: How do you configure parallel jobs in Azure Pipelines?

A: Purchase additional parallel jobs in Organization Settings > Parallel jobs. Each parallel job allows one build/release to run simultaneously.

Question 40

Q: What is the purpose of task groups in Azure Pipelines?

A: Task groups encapsulate a sequence of tasks as a single reusable unit, allowing consistent task configurations across multiple pipelines.

Question 41

Q: How do you implement rolling deployment?

A: Use deployment groups with the "Rolling" deployment strategy, specifying batch size to deploy to a subset of machines at a time while others remain available.

Question 42

Q: What are YAML pipeline conditions used for?

A: Conditions control whether jobs, stages, or steps run based on expressions evaluating variables, previous results, or custom logic using `condition:` keyword.

Question 43

Q: How do you publish build artifacts?

A: Use the PublishBuildArtifacts task or `publish:` shorthand in YAML, specifying the path to publish and artifact name for downstream consumption.

Question 44

Q: What is the Publish Code Coverage Results task used for?

A: This task publishes code coverage results (Cobertura, JaCoCo format) to Azure Pipelines, displaying coverage metrics and reports in the pipeline summary.

Question 45

Q: How do you configure pipeline triggers for pull requests?

A: Use `pr:` trigger in YAML to run pipelines on PR creation/updates. Specify branches to target and optionally filter by paths or draft status.

Question 46

Q: What is the purpose of service connections?

A: Service connections store credentials and endpoints for external services (Azure, Docker registries, Kubernetes, GitHub), enabling secure access from pipelines.

Question 47

Q: How do you implement infrastructure as code with ARM templates?

A: Use the Azure Resource Group Deployment task in pipelines, providing ARM template and parameter files to provision Azure resources declaratively.

Question 48

Q: What is Bicep and how does it relate to ARM templates?

A: Bicep is a domain-specific language that compiles to ARM templates, offering simpler syntax, modules, and better tooling while deploying the same infrastructure.

Question 49

Q: How do you configure Terraform deployments in Azure Pipelines?

A: Install the Terraform extension, configure backend storage for state, then use Terraform tasks for init, plan, and apply commands in your pipeline.

Question 50

Q: What is the purpose of environments in YAML pipelines?

A: Environments represent deployment targets (dev, staging, prod) with approvals, checks, and deployment history tracking for better governance and traceability.

Question 51

Q: How do you implement container deployments to AKS?

A: Build and push images to Azure Container Registry, then use Kubectl, Helm, or Azure Kubernetes Service tasks to deploy manifests or charts to AKS clusters.

Question 52

Q: What is Helm and how is it used in Azure Pipelines?

A: Helm is a Kubernetes package manager using charts to define, install, and upgrade applications. Use Helm tasks to package and deploy charts to clusters.

Question 53

Q: How do you configure automatic triggers on scheduled intervals?

A: Use `schedules:` in YAML with cron expressions to trigger pipelines at specific times. Specify branches and whether to run only when source has changed.

Question 54

Q: What is the purpose of the Checkout task?

A: The Checkout task clones the repository code to the agent. Configure shallow fetch, submodule checkout, and clean options for optimized source retrieval.

Question 55

Q: How do you pass variables between pipeline stages?

A: Use output variables with `isOutput=true`, then reference in dependent stages using `stageDependencies.stageName.jobName.outputs['stepName.variableName']`.

Question 56

Q: What is pipeline resource triggers?

A: Resource triggers start pipelines when upstream pipelines, container images, or packages are updated, enabling automated downstream builds and deployments.

Question 57

Q: How do you implement feature flags with LaunchDarkly?

A: Install the LaunchDarkly extension, add feature flag checks in code, use LaunchDarkly tasks in pipelines to enable/disable flags during deployment.

Question 58

Q: What is the purpose of the Azure CLI task?

A: The Azure CLI task runs Azure CLI commands authenticated to a subscription via service connection, enabling custom Azure resource management in pipelines.

Question 59

Q: How do you configure artifact retention policies?

A: In Project Settings > Pipelines > Settings, configure retention days for runs, releases, and artifacts. Override at pipeline level using retention rules.

Question 60

Q: What is a Universal Package in Azure Artifacts?

A: Universal Packages store any file types (binaries, tools, datasets) with versioning, allowing sharing of non-standard artifacts across projects and pipelines.

Question 61

Q: How do you implement database deployments in pipelines?

A: Use tools like SQL Server Data Tools (SSDT), Entity Framework migrations, or Flyway with appropriate tasks to apply schema changes and migrations during deployment.

Question 62

Q: What is the purpose of deployment jobs in YAML?

A: Deployment jobs target environments with deployment strategies (runOnce, rolling, canary), enabling environment-specific approvals and deployment history.

Question 63

Q: How do you configure self-hosted agent pools?

A: Create agent pools in Organization Settings, download and configure agents on your machines, register with PAT, then reference pool name in pipelines.

Question 64

Q: What is the batch trigger in Azure Pipelines?

A: `batch: true` in triggers batches multiple commits into single runs, preventing pipeline queue buildup when many commits occur rapidly.

Question 65

Q: How do you implement test automation in pipelines?

A: Add test tasks (VSTest, dotnet test, npm test) to pipelines, configure test result publishing for reporting, and implement quality gates based on results.

Question 66

Q: What is the purpose of the Azure Web App Deploy task?

A: This task deploys applications to Azure App Service, supporting deployment slots, configuration, and various source types (package, folder, container).

Question 67

Q: How do you configure webhook triggers?

A: Create incoming webhooks in pipelines or use Service Hooks to trigger pipelines from external services via HTTP POST requests with authentication.

Question 68

Q: What is the difference between jobs and stages in YAML?

A: Stages are collections of jobs that run sequentially with dependencies. Jobs run on agents and can run in parallel within a stage.

Question 69

Q: How do you implement load testing in pipelines?

A: Use Azure Load Testing service with the Azure Load Testing task, or integrate third-party tools like JMeter, to run performance tests during deployment.

Question 70

Q: What is the purpose of pipeline variables?

A: Variables store values reused across pipeline steps. Define at pipeline, stage, job, or step level. Use variable groups for shared values across pipelines.

Question 71

Q: How do you secure pipeline variables?

A: Mark variables as secret to mask in logs. Store secrets in Azure Key Vault linked to variable groups. Use minimal permissions for service connections.

Question 72

Q: What is the extends keyword in YAML pipelines?

A: `extends:` allows pipelines to inherit from templates, enabling centralized control of pipeline structure while allowing customization of parameters.

Question 73

Q: How do you configure matrix builds?

A: Use `strategy: matrix:` in jobs to run the same job with different variable combinations (e.g., multiple OS versions or Node versions) in parallel.

Question 74

Q: What is the purpose of pipeline decorators?

A: Pipeline decorators are extensions that automatically inject tasks into pipelines organization-wide, ensuring consistent security scanning or compliance checks.

Question 75

Q: How do you implement container jobs?

A: Specify `container:` in jobs to run steps inside a container image, providing consistent environments without agent configuration dependencies.

Domain 4: Develop a Security and Compliance Plan (10-15%)

Question 76

Q: What is the difference between Service Principals and Managed Identities?

A: Service Principals are Azure AD app registrations with credentials you manage. Managed Identities are Azure-managed identities for resources eliminating credential management.

Question 77

Q: How do you implement static code analysis in pipelines?

A: Add SonarQube/SonarCloud tasks for code quality and security scanning. Configure quality gates to fail builds on critical issues or insufficient coverage.

Question 78

Q: What is the purpose of WhiteSource Bolt/Mend?

A: WhiteSource (now Mend) scans for open-source vulnerabilities and license compliance, identifying security risks in third-party dependencies.

Question 79

Q: How do you configure SonarQube in Azure Pipelines?

A: Install SonarQube extension, configure service connection to SonarQube server, add Prepare, Run, and Publish tasks around build steps.

Question 80

Q: What is OWASP ZAP used for?

A: OWASP ZAP performs dynamic application security testing (DAST), scanning running applications for vulnerabilities like SQL injection and XSS.

Question 81

Q: How do you implement secret scanning in repositories?

A: Enable GitHub Advanced Security or Azure DevOps secret scanning to detect accidentally committed credentials, API keys, and tokens in code.

Question 82

Q: What is the purpose of Azure Policy in DevOps?

A: Azure Policy enforces organizational standards on Azure resources, ensuring deployments comply with security, naming, and configuration requirements.

Question 83

Q: How do you configure repository permissions in Azure Repos?

A: Navigate to Project Settings > Repositories, configure security for branches and repos, setting allow/deny for read, contribute, manage permissions.

Question 84

Q: What are personal access tokens (PATs) used for?

A: PATs provide scoped authentication for Azure DevOps APIs, Git operations, and integrations. Configure specific permissions and expiration dates for security.

Question 85

Q: How do you implement credential scanning with CredScan?

A: Add the Credential Scanner task (Microsoft Security DevOps) to pipelines to detect hardcoded credentials, blocking builds with security violations.

Question 86

Q: What is GitHub Dependabot?

A: Dependabot automatically creates PRs to update vulnerable dependencies, keeping projects secure by monitoring for known security advisories.

Question 87

Q: How do you configure branch protection rules?

A: Enable branch policies requiring PR reviews, build validation, linked work items, and enforce permissions to prevent direct pushes to protected branches.

Question 88

Q: What is the principle of least privilege in DevOps?

A: Grant minimum necessary permissions for users and service accounts. Use scoped tokens, specific roles, and regular access reviews.

Question 89

Q: How do you implement container image scanning?

A: Enable Microsoft Defender for Container Registries or use tools like Trivy, Twistlock, or Aqua to scan images for vulnerabilities before deployment.

Question 90

Q: What is the purpose of Azure Security Center in DevOps?

A: Azure Security Center provides security recommendations, vulnerability assessments, and compliance monitoring for Azure resources and workloads.

Question 91

Q: How do you configure GITHUB_TOKEN permissions?

A: In workflow files, use `permissions:` block to restrict GITHUB_TOKEN scope. Set default permissions in repository settings to read-only minimum.

Question 92

Q: What is software composition analysis (SCA)?

A: SCA analyzes third-party components and dependencies for known vulnerabilities, license compliance, and outdated versions requiring updates.

Question 93

Q: How do you implement governance with initiative policies?

A: Create Azure Policy initiatives grouping related policies, assign to management groups or subscriptions for consistent compliance across resources.

Question 94

Q: What is the purpose of audit logs in Azure DevOps?

A: Audit logs track user actions, permission changes, and administrative events for security monitoring, compliance, and incident investigation.

Question 95

Q: How do you secure pipeline service connections?

A: Restrict service connection access to specific pipelines, use managed identities where possible, and implement connection check approvals for production.

Question 96

Q: What is threat modeling in DevSecOps?

A: Threat modeling identifies potential security threats during design phase, analyzing attack surfaces and implementing mitigations before development.

Question 97

Q: How do you implement compliance as code?

A: Define compliance policies in Azure Policy or OPA, scan infrastructure code for compliance, and enforce through pipeline gates and deployment checks.

Question 98

Q: What is the purpose of GitHub code scanning?

A: GitHub code scanning uses CodeQL to analyze code for security vulnerabilities and coding errors, integrating results into pull requests.

Question 99

Q: How do you configure required reviewers for sensitive paths?

A: Use CODEOWNERS files (GitHub) or path-based policies (Azure DevOps) to automatically require approval from specific teams for changes to sensitive files.

Question 100

Q: What is Azure DevOps organization-level security?

A: Organization settings control member access, external guest access, OAuth applications, and security policies applying to all projects within the organization.

Domain 5: Implement an Instrumentation Strategy (5-10%)

Question 101

Q: What is the purpose of Azure Application Insights?

A: Application Insights provides application performance monitoring (APM), collecting telemetry on requests, dependencies, exceptions, and user behavior.

Question 102

Q: How do you configure Application Insights for a web application?

A: Add the Application Insights SDK, configure the instrumentation key or connection string, and deploy. Auto-instrumentation is available for some platforms.

Question 103

Q: What is the purpose of Azure Monitor?

A: Azure Monitor collects and analyzes metrics and logs from Azure resources, providing alerting, visualization, and integration with monitoring tools.

Question 104

Q: How do you create alerts based on application metrics?

A: In Azure Monitor, create alert rules with conditions on metrics or log queries. Configure action groups for notifications via email, SMS, webhooks, or automation.

Question 105

Q: What is distributed tracing in Application Insights?

A: Distributed tracing tracks requests across microservices using correlation IDs, visualizing dependencies and identifying performance bottlenecks.

Question 106

Q: How do you implement custom telemetry?

A: Use Application Insights SDK to track custom events, metrics, and properties with TrackEvent, TrackMetric, and TrackTrace methods in application code.

Question 107

Q: What is the Application Map in Application Insights?

A: Application Map visualizes application components and their dependencies, showing request flows, failure rates, and performance across services.

Question 108

Q: How do you configure log analytics workspaces?

A: Create Log Analytics workspaces in Azure, configure diagnostic settings to send logs from resources, and use KQL queries for analysis and alerts.

Question 109

Q: What are availability tests in Application Insights?

A: Availability tests ping application endpoints from multiple locations, alerting when sites become unavailable or response times exceed thresholds.

Question 110

Q: How do you implement monitoring dashboards?

A: Create Azure dashboards with metrics charts, log query visualizations, and Application Insights widgets. Use Azure Workbooks for interactive reports.

Question 111

Q: What is Smart Detection in Application Insights?

A: Smart Detection automatically detects performance anomalies and potential issues using machine learning, notifying teams of unusual patterns.

Question 112

Q: How do you configure live metrics stream?

A: Live Metrics Stream shows real-time telemetry with 1-second granularity, useful for monitoring deployments and debugging live issues.

Question 113

Q: What is the purpose of Azure Monitor Logs?

A: Azure Monitor Logs stores log data in Log Analytics workspaces, enabling complex queries, visualizations, and alerts using Kusto Query Language (KQL).

Question 114

Q: How do you implement end-to-end transaction monitoring?

A: Configure Application Insights with dependency tracking, use correlation headers, and analyze Transaction diagnostics view for complete request flows.

Question 115

Q: What is App Center used for?

A: Visual Studio App Center provides mobile app analytics, crash reporting, distribution, and CI/CD for iOS, Android, and Windows applications.

Question 116

Q: How do you configure diagnostic settings for Azure resources?

A: In resource settings, configure Diagnostic settings to send platform logs and metrics to Log Analytics, Storage accounts, or Event Hubs.

Question 117

Q: What are Azure Monitor metrics?

A: Metrics are numerical values collected at regular intervals describing resource aspects. Platform metrics are automatic; custom metrics can be published via SDK.

Question 118

Q: How do you implement user behavior analytics?

A: Use Application Insights Users, Sessions, Events views. Configure User Flows to analyze navigation paths and Funnels for conversion tracking.

Question 119

Q: What is the purpose of Azure Service Health?

A: Azure Service Health provides personalized alerts and guidance for Azure service issues, planned maintenance, and health advisories affecting your resources.

Question 120

Q: How do you configure alerts for pipeline failures?

A: Configure Service Hooks or notifications in Azure DevOps to send alerts to Slack, Teams, email, or webhooks when builds or releases fail.

Additional Practice Questions

Question 121

Q: What is SemVer and how is it applied in Azure Artifacts?

A: Semantic Versioning uses MAJOR.MINOR.PATCH format. Increment MAJOR for breaking changes, MINOR for features, PATCH for bug fixes. Azure Artifacts supports SemVer for packages.

Question 122

Q: How do you configure upstream sources in Azure Artifacts?

A: In feed settings, add upstream sources (nuget.org, npmjs.com) to cache public packages, reducing external dependencies and improving reliability.

Question 123

Q: What is the purpose of feed views in Azure Artifacts?

A: Views (like @release, @prerelease) filter which package versions consumers see, enabling promotion workflows from development to release feeds.

Question 124

Q: How do you implement NuGet package restoration in pipelines?

A: Use NuGet restore or dotnet restore task, configure authentication for private feeds using nuget.config or credential provider.

Question 125

Q: What is the purpose of deployment slots in Azure App Service?

A: Deployment slots host different app versions with separate configurations, enabling warm-up, testing, and zero-downtime swaps to production.

Question 126

Q: How do you configure auto-scaling for Azure resources?

A: Define autoscale settings with rules based on metrics (CPU, memory, queue depth) to automatically increase or decrease resource instances.

Question 127

Q: What is Azure Resource Manager (ARM)?

A: ARM is Azure's deployment and management service, providing consistent management layer for creating, updating, and organizing Azure resources.

Question 128

Q: How do you implement linked ARM templates?

A: Reference external templates using linkedTemplate in ARM, enabling modular, reusable infrastructure components deployed together.

Question 129

Q: What is the purpose of parameter files in ARM templates?

A: Parameter files provide environment-specific values for ARM template parameters, enabling the same template to deploy to different environments.

Question 130

Q: How do you configure Desired State Configuration (DSC)?

A: Create DSC configurations defining desired server state, compile to MOF files, and apply using Azure Automation State Configuration or local LCM.

Question 131

Q: What is Azure Automation used for in DevOps?

A: Azure Automation provides runbooks, DSC, update management, and process automation for consistent configuration and operational tasks.

Question 132

Q: How do you implement configuration management with Chef or Puppet?

A: Use Chef cookbooks or Puppet manifests to define configuration, integrate with pipelines using extensions, and apply to target nodes via agents.

Question 133

Q: What is the purpose of Azure Container Registry (ACR)?

A: ACR provides private Docker registry for storing and managing container images with geo-replication, security scanning, and Azure AD integration.

Question 134

Q: How do you implement container image tagging strategies?

A: Use meaningful tags combining version numbers, git commits, and build IDs. Avoid relying solely on ‘latest’ tag for production deployments.

Question 135

Q: What is AKS and how is it used for deployments?

A: Azure Kubernetes Service (AKS) is managed Kubernetes for container orchestration. Deploy using kubectl, Helm, or Azure DevOps Kubernetes tasks.

Question 136

Q: How do you configure Kubernetes namespaces for environments?

A: Create separate namespaces (dev, staging, prod) for isolation. Apply resource quotas, network policies, and RBAC per namespace.

Question 137

Q: What is a ConfigMap in Kubernetes?

A: ConfigMaps store non-confidential configuration data as key-value pairs, consumed by pods as environment variables or mounted files.

Question 138

Q: How do you manage secrets in Kubernetes?

A: Use Kubernetes Secrets for sensitive data, integrate with Azure Key Vault using CSI driver, or use external secret management solutions.

Question 139

Q: What is a Kubernetes Deployment resource?

A: Deployment manages ReplicaSets and provides declarative updates for pods, handling rolling updates, rollbacks, and scaling.

Question 140

Q: How do you implement health checks in Kubernetes?

A: Configure liveness probes (restart unhealthy containers) and readiness probes (control traffic routing) in pod specifications.

Question 141

Q: What is the purpose of GitHub Actions?

A: GitHub Actions provides CI/CD workflows triggered by repository events, using YAML workflow files for building, testing, and deploying applications.

Question 142

Q: How do you configure GitHub Actions secrets?

A: Store secrets in repository or organization settings. Reference in workflows using ${{ secrets.SECRET_NAME }} syntax. Secrets are encrypted.

Question 143

Q: What is the difference between GitHub Actions and Azure Pipelines?

A: Both provide CI/CD. GitHub Actions is native to GitHub with marketplace actions. Azure Pipelines offers deeper Azure integration and more hosting options.

Question 144

Q: How do you implement workflow dispatch triggers?

A: Add `workflow_dispatch:` to triggers, enabling manual pipeline runs with optional input parameters from the GitHub Actions UI.

Question 145

Q: What is a composite action in GitHub Actions?

A: Composite actions combine multiple steps into a reusable action defined in action.yml, reducing duplication across workflows.

Question 146

Q: How do you configure conditional step execution in GitHub Actions?

A: Use `if:` conditions with expressions checking job status, variables, or contexts. Example: `if: success()` or `if: github.ref == 'refs/heads/main'`

Question 147

Q: What is the purpose of the actions/checkout action?

A: actions/checkout clones the repository to the runner, enabling subsequent steps to access source code. Configure depth, submodules, and ref options.

Question 148

Q: How do you implement artifact sharing between jobs?

A: Use actions/upload-artifact and actions/download-artifact to share files between jobs in the same workflow.

Question 149

Q: What are GitHub Apps used for?

A: GitHub Apps provide API access with granular permissions, acting on behalf of the app. Used for integrations, bots, and automation tools.

Question 150

Q: How do you configure required workflows in GitHub?

A: Enterprise and organization admins can require specific workflows to run on matching repositories, ensuring consistent checks across projects.

Question 151

Q: What is the purpose of test flakiness detection?

A: Flaky tests are unreliable tests that pass and fail intermittently. Azure DevOps tracks test flakiness to identify and address unstable tests.

Question 152

Q: How do you implement unit testing in pipelines?

A: Add test execution tasks (dotnet test, pytest, npm test) and publish results using test result publishing tasks for visibility in pipeline UI.

Question 153

Q: What is the difference between unit and integration tests?

A: Unit tests verify individual components in isolation. Integration tests verify interactions between components, often requiring deployed services or databases.

Question 154

Q: How do you configure test agents for distributed testing?

A: Set up test controller and agents for load testing, or use Azure Test Plans cloud-based test agents for executing automated tests.

Question 155

Q: What is the purpose of test impact analysis?

A: Test impact analysis identifies which tests are affected by code changes, optimizing test execution by running only relevant tests.

Question 156

Q: How do you implement manual testing with Azure Test Plans?

A: Create test cases in Azure Test Plans, organize in test suites, execute tests using web-based Test Runner, and track results and defects.

Question 157

Q: What is exploratory testing?

A: Exploratory testing is unscripted testing where testers explore application functionality, using Azure Test Plans extension to capture findings.

Question 158

Q: How do you configure quality gates based on test results?

A: Set minimum pass percentage or coverage thresholds in pipeline tasks or release gates. Fail builds or block deployments on quality failures.

Question 159

Q: What is the purpose of code coverage metrics?

A: Code coverage measures how much code is executed by tests, identifying untested areas. Use tools like Cobertura or JaCoCo to collect coverage.

Question 160

Q: How do you implement A/B testing?

A: Use feature flags or traffic splitting (Azure Front Door, App Configuration) to route users to different versions and measure outcomes.

Question 161

Q: What is GitVersion used for?

A: GitVersion automatically generates version numbers based on Git history and branching strategy, implementing semantic versioning in pipelines.

Question 162

Q: How do you configure build versioning?

A: Use build numbers with format strings, GitVersion tasks, or custom scripts to generate consistent, meaningful version numbers.

Question 163

Q: What is the purpose of artifact versioning?

A: Artifact versioning ensures unique, traceable package versions. Use build numbers, git commits, or semantic versioning for packages.

Question 164

Q: How do you implement package promotion workflows?

A: Use feed views in Azure Artifacts. Packages start in @prerelease, are promoted to @release after validation, controlling what consumers receive.

Question 165

Q: What is the difference between release and deployment?

A: A release is a versioned artifact set approved for deployment. Deployment is the process of installing the release to an environment.

Question 166

Q: How do you configure release pipelines vs YAML pipelines?

A: Release pipelines use visual designer for deployments. YAML pipelines define everything as code. Both support environments, approvals, and gates.

Question 167

Q: What is progressive exposure deployment?

A: Progressive exposure gradually rolls out changes to increasing user percentages using rings (internal users, early adopters, all users).

Question 168

Q: How do you implement rollback strategies?

A: Maintain previous deployments for quick rollback. Use deployment slots, blue-green, or automated rollback on health check failures.

Question 169

Q: What is the purpose of deployment rings?

A: Deployment rings group users by risk tolerance. Deploy to inner rings (canary) first, progressively expanding to outer rings after validation.

Question 170

Q: How do you configure traffic manager for deployments?

A: Azure Traffic Manager routes traffic across deployments using weighted routing for gradual rollout or priority routing for failover.

Question 171

Q: What is Azure Front Door?

A: Azure Front Door provides global load balancing, SSL termination, and traffic routing with rules-based routing and WAF for applications.

Question 172

Q: How do you implement dark launching?

A: Deploy features to production in disabled state using feature flags. Enable for testing without exposing to all users.

Question 173

Q: What is the purpose of the Pre-deployment conditions?

A: Pre-deployment conditions specify approvals, gates, and triggers that must be satisfied before deployment to a stage begins.

Question 174

Q: How do you configure post-deployment gates?

A: Add gates after deployment to verify health metrics, run smoke tests, or check external systems before considering deployment successful.

Question 175

Q: What is the purpose of Azure Boards queries?

A: Queries find and display work items matching criteria. Use for reporting, dashboards, and release gates checking for open bugs.

Question 176

Q: How do you link deployments to work items?

A: Enable work item integration in release settings. Azure DevOps automatically links deployed work items and updates their state.

Question 177

Q: What is the purpose of deployment groups vs environments?

A: Deployment groups target on-premises VMs with agents. YAML environments provide approvals and tracking for any deployment target.

Question 178

Q: How do you configure VM deployments?

A: Use deployment groups with agents on VMs, or use Azure Resource Manager tasks to manage VM resources and extensions.

Question 179

Q: What is the purpose of the Azure DevOps REST API?

A: REST APIs provide programmatic access to Azure DevOps services for automation, integration, and custom tooling beyond UI capabilities.

Question 180

Q: How do you automate pipeline creation?

A: Use REST APIs or Azure CLI to create/update pipelines programmatically. Store pipeline definitions in repos for GitOps-style management.

Question 181

Q: What is the purpose of compliance scanning in pipelines?

A: Compliance scanning verifies infrastructure and code meet regulatory requirements (PCI, HIPAA, SOC2) using policy-as-code tools.

Question 182

Q: How do you implement security scanning for containers?

A: Scan images with Microsoft Defender for Cloud, Trivy, or Aqua before and after pushing to registry. Fail pipelines on critical vulnerabilities.

Question 183

Q: What is the Open Policy Agent (OPA)?

A: OPA is a policy engine for unified policy enforcement across the stack, enabling policy-as-code for Kubernetes, APIs, and infrastructure.

Question 184

Q: How do you implement Kubernetes admission control?

A: Use admission controllers like OPA Gatekeeper to validate and mutate resources, enforcing policies on pod security, labels, and images.

Question 185

Q: What is the purpose of Azure Defender for DevOps?

A: Azure Defender for DevOps provides security posture management for DevOps environments, identifying misconfigurations and vulnerabilities across pipelines.

Question 186

Q: How do you secure Azure DevOps extensions?

A: Review extension permissions before installation, use only trusted publishers, and audit installed extensions regularly.

Question 187

Q: What is the purpose of service principals in Azure DevOps?

A: Service principals authenticate Azure resource access from pipelines without user credentials. Configure with certificates or secrets in service connections.

Question 188

Q: How do you implement just-in-time access?

A: Use Azure AD Privileged Identity Management (PIM) to grant temporary elevated permissions for administrative tasks when needed.

Question 189

Q: What is the purpose of Azure DevOps Security scanning?

A: Built-in security scanning identifies vulnerabilities in code and dependencies. Enable Advanced Security for repository-level scanning.

Question 190

Q: How do you configure multi-repo triggers?

A: In YAML, use resources.repositories to reference additional repositories and trigger pipelines when changes occur in any configured repo.

Question 191

Q: What is the purpose of pipeline runs retention?

A: Retention policies automatically delete old pipeline runs to manage storage. Configure retention days and keep minimum successful/failed runs.

Question 192

Q: How do you implement infrastructure testing?

A: Use tools like Terraform test, ARM TTK, or Pester to validate infrastructure code before deployment. Include in pipeline validation stages.

Question 193

Q: What is chaos engineering?

A: Chaos engineering proactively tests system resilience by introducing failures (Azure Chaos Studio) to identify weaknesses before production issues.

Question 194

Q: How do you configure pipeline analytics?

A: View pipeline analytics in Azure DevOps for build durations, failure rates, and trends. Use Analytics views for custom reports.

Question 195

Q: What is the purpose of runtime parameters in YAML?

A: Runtime parameters accept values when queuing pipelines, enabling dynamic configuration without modifying pipeline code.

Question 196

Q: How do you implement secrets rotation?

A: Store secrets in Key Vault with expiration dates, use managed identities where possible, and automate rotation with Azure Functions or Logic Apps.

Question 197

Q: What is the purpose of Dependabot alerts?

A: Dependabot alerts notify when repository dependencies have known security vulnerabilities, providing remediation guidance.

Question 198

Q: How do you configure automated pull request creation?

A: Use Dependabot for security updates, Azure DevOps REST API, or GitHub Actions to automatically create PRs for dependency updates.

Question 199

Q: What is the purpose of workload identity federation?

A: Workload identity federation enables Azure AD authentication from external identity providers (GitHub, GitLab) without managing secrets.

Question 200

Q: How do you implement GitOps for Kubernetes?

A: Use Flux or Argo CD to synchronize Kubernetes state with Git repositories, automatically applying changes when repository is updated.

Repost and Follow

Nensi Ravaliya

for more content

Want to build your career in cloud?

Subscribe to Yatri Cloud Channel