Google GCP-PCDE

Google Professional Cloud DevOps Engineer Certification Questions & Answers

Get Instant Access to Vital Exam Acing Materials | Study Guide | Sample Questions | Practice Test

GCP-PCDE

Google Cloud Platform - Professional Cloud DevOps Engineer (GCP-PCDE)

50 Questions Exam – 70% Cut Score – Duration of 120 minutes

Table of Contents:

Discover More about the GCP-PCDE Certification

Are you interested in passing the Google GCP-PCDE exam? First discover, who benefits from the GCP-PCDE certification. The GCP-PCDE is suitable for a candidate if he wants to learn about Cloud. Passing the GCP-PCDE exam earns you the Google Cloud Platform - Professional Cloud DevOps Engineer (GCP-PCDE) title.

While preparing for the GCP-PCDE exam, many candidates struggle to get the necessary materials. But do not worry; your struggling days are over. The GCP-PCDE PDF contains some of the most valuable preparation tips and the details and instant access to useful **GCP-PCDE study materials just at one click**.

Google GCP-PCDE Professional Cloud DevOps Engineer Certification Details:

Exam Name	Google Professional Cloud DevOps Engineer (GCP-PCDE)
Exam Code	GCP-PCDE
Exam Price	$200 USD
Duration	120 minutes
Number of Questions	50
Passing Score	Pass / Fail (Approx 70%)
Recommended Training / Books	**Google Cloud documentation** **Google Cloud solutions**
Schedule Exam	**Google Cloud Webassessor**
Sample Questions	**Google GCP-PCDE Sample Questions**
Recommended Practice	**Google Cloud Platform - Professional Cloud DevOps Engineer (GCP-PCDE) Practice Test**

GCP-PCDE Syllabus:

Section	**Objectives**
Applying site reliability engineering principles to a service	
Balance change, velocity, and reliability of the service:	- Discover SLIs (e.g., availability, latency) - Define SLOs and understand SLAs - Agree to consequences of not meeting the error budget - Construct feedback loops to decide what to build next - Eliminate toil via automation

Section	Objectives
Manage service life cycle:	- Manage a service (e.g., introduce a new service, deploy, maintain, and retire it) - Plan for capacity (e.g., quotas and limits management)
Ensure healthy communication and collaboration for operations:	- Prevent burnout (e.g., set up automation processes to prevent burnout) - Foster a learning culture - Foster a culture of blamelessness
Building and implementing CI/CD pipelines for a service	
Design CI/CD pipelines:	- Creating and storing immutable artifacts with Artifact Registry - Deployment strategies with Cloud Build and Spinnaker - Deployment to hybrid and multicloud environments with Anthos, Spinnaker, and Kubernetes - Artifact versioning strategy with Cloud Build and Artifact Registry - CI/CD pipeline triggers with Cloud Source Repositories, external SCM, and Pub/Sub - Testing a new version with Spinnaker - Configuring deployment processes (e.g., approval flows)
Implement CI/CD pipelines:	- CI with Cloud Build - CD with Cloud Build - Open source tooling (e.g., Jenkins, Spinnaker, GitLab, Concourse) - Auditing and tracing of deployments (e.g., CSR, Artifact Registry, Cloud Build, Cloud Audit Logs)
Manage configuration and secrets:	- Secure storage methods - Secret rotation and config changes
Manage infrastructure as code:	- Terraform - Infrastructure code versioning - Make infrastructure changes safer - Immutable architecture
Deploy CI/CD tooling:	- Centralized tools vs. multiple tools (single vs. multi-tenant) - Security of CI/CD tooling
Manage different development environments (e.g., staging, production):	- Decide on the number of environments and their purpose - Create environments dynamically per feature branch with GKE - Local development environments with Docker, Cloud Code, Skaffold
Secure the deployment pipeline:	- Vulnerability analysis with Artifact Registry - Binary Authorization - IAM policies per environment

Section	Objectives
Implementing service monitoring strategies	
Manage application logs:	- Collecting logs from Compute Engine, GKE with Cloud Logging, Fluentd - Collecting third-party and structured logs with Cloud Logging, Fluentd - Sending application logs directly to the Cloud Logging API
Manage application metrics with Cloud Monitoring:	- Collecting metrics from Compute Engine - Collecting GKE/Kubernetes metrics - Use Metrics Explorer for ad hoc metric analysis
Manage Cloud Monitoring platform:	- Creating a monitoring dashboard - Filtering and sharing dashboards - Configure third-party alerting in Cloud Monitoring (e.g., PagerDuty, Slack) - Define alerting policies based on SLIs with Cloud Monitoring - Automate alerting policy definition with Terraform - Implementing SLO monitoring and alerting with Cloud Monitoring - Understand Cloud Monitoring integrations (e.g., Grafana, BigQuery) - Using SIEM tools to analyze audit/flow logs (e.g., Splunk, Datadog) - Design Cloud Monitoring metrics scopes
Manage Cloud Logging platform:	- Enabling data access logs (e.g., Cloud Audit Logs) - Enabling VPC flow logs - Viewing logs in the Google Cloud Console - Using basic vs. advanced logging filters - Implementing logs-based metrics - Understanding the logging exclusion vs. logging export - Selecting the options for logging export - Implementing a project-level / org-level export - Viewing export logs in Cloud Storage and BigQuery - Sending logs to an external logging platform
Implement logging and monitoring access controls:	- Set ACL to restrict access to audit logs with IAM, Cloud Logging - Set ACL to restrict export configuration with IAM, Cloud Logging - Set ACL to allow metric writing for custom metrics with IAM, Cloud Monitoring
Optimizing service performance	
Identify service performance issues:	- Evaluate and understand user impact - Utilize Google Cloud's operations suite to identify cloud

Section	Objectives
	resource utilization - Utilize Cloud Trace and Cloud Profiler to profile performance characteristics - Interpret service mesh telemetry - Troubleshoot issues with the image/OS - Troubleshoot network issues (e.g., VPC flow logs, firewall logs, latency, view network details)
Debug application code:	- Application instrumentation - Cloud Debugger - Cloud Logging - Cloud Trace - Debugging distributed applications - App Engine local development server - Error Reporting - Cloud Profiler
Optimize resource utilization:	- Identify resource costs - Identify resource utilization levels - Develop plan to optimize areas of greatest cost or lowest utilization - Manage preemptible VMs - Utilize committed use discounts where appropriate - TCO considerations (e.g., security, logging, networking) - Consider network pricing
Managing service incidents	
Coordinate roles and implement communication channels during a service incident:	- Define roles (incident commander, communication lead, operations lead) - Handle requests for impact assessment - Provide regular status updates, internal and external - Record major changes in incident state (e.g., When mitigated? When is all clear?) - Establish communications channels (e.g., email, IRC, Hangouts, Slack, phone) - Scaling response team and delegation - Avoid exhaustion / burnout - Rotate / hand over roles - Manage stakeholder relationships
Investigate incident symptoms impacting users:	- Identify probable causes of service failure - Evaluate symptoms against probable causes; rank probability of cause based on observed behavior - Perform investigation to isolate most likely actual cause - Identify alternatives to mitigate issue
Mitigate incident impact on users:	- Roll back release - Drain / redirect traffic

Section	Objectives
	- Turn off experiment - Add capacity
Resolve issues with deployments (e.g., Cloud Build, Jenkins):	- Code change / fix bug - Verify fix - Declare all-clear
Document issue in a postmortem:	- Document root causes - Create and prioritize action items - Communicate postmortem to stakeholders

Broaden Your Knowledge with Google GCP-PCDE Sample Questions:

Question: 1

You have a Compute Engine instance that uses the default Debian image. The application hosted on this instance recently suffered a series of crashes that you weren't able to debug in real time: the application process died suddenly every time.

The application usually consumes 50% of the instance's memory, and normally never more than 70%, but you suspect that a memory leak was responsible for the crashes. You want to validate this hypothesis.

What should you do?

a) Go to Metrics Explorer and look for the "compute.googleapis.com/guest/system/problem_count" metric for that instance. Examine its value for when the application crashed in the past.
b) In Cloud Monitoring, create an uptime check for your application. Create an alert policy for that uptime check to be notified when your application crashes. When you receive an alert, use your usual debugging tools to investigate the behavior of the application in real time.
c) Install the Cloud Monitoring agent on the instance. Go to Metrics Explorer and look for the "agent.googleapis.com/memory/percent_used" metric for that instance. Examine its value for when the application crashed in the past.
d) Install the Cloud Monitoring agent on the instance. Create an alert policy on the "agent.googleapis.com/memory/percent_used" metric for that instance to be alerted when the memory used is higher than 75%. When you receive an alert, use your usual debugging tools to investigate the behavior of the application in real time.

Answer: d

Question: 2

You are deploying an application to a Kubernetes cluster that requires a username and password to connect to another service.

When you deploy the application, you want to ensure that the credentials are used securely in multiple environments with minimal code changes.

What should you do?

a) Bundle the credentials with the code inside the container and secure the container registry.
b) Leverage a CI/CD pipeline to update the variables at build time and inject them into a templated Kubernetes application manifest.
c) Store the credentials as a Kubernetes Secret and let the application access it via environment variables at runtime.
d) Store the credentials as a Kubernetes ConfigMap and let the application access it via environment variables at runtime.

Answer: c

Question: 3

Several teams in your company want to use Cloud Build to deploy to their own Google Kubernetes Engine (GKE) clusters.

The clusters are in projects that are dedicated to each team. The teams only have access to their own projects. One team should not have access to the cluster of another team.

You are in charge of designing the Cloud Build setup, and want to follow Google-recommended practices. What should you do?

a) Limit each team member's access so that they only have access to their team's clusters. Ask each team member to install the gcloud CLI and to authenticate themselves by running “gcloud init”. Ask each team member to execute Cloud Build builds by using “gcloud builds submit”.
b) Create a single project for Cloud Build that all the teams will use. List the service accounts in this project and identify the one used by Cloud Build. Grant the Kubernetes Engine Developer IAM role to that service account in each team's project.
c) In each team's project, list the service accounts and identify the one used by Cloud Build for each project. In each project, grant the Kubernetes Engine Developer IAM role to the service account used by Cloud Build. Ask each team to execute Cloud Build builds in their own project.
d) In each team's project, create a service account, download a JSON key for that service account, and grant the Kubernetes Engine Developer IAM role to that service account in that project. Create a single project for Cloud Build that all the teams will use. In that project, encrypt all the service account keys by using Cloud KMS. Grant the Cloud KMS CryptoKey Decrypter IAM role to Cloud Build's service account. Ask each team to include in their “cloudbuild.yaml” files a step that decrypts the key of their service account, and use that key to connect to their cluster.

Answer: c

Question: 4

You work with a video rendering application that publishes small tasks as messages to a Cloud Pub/Sub topic. You need to deploy the application that will execute these tasks on multiple virtual machines (VMs).

Each task takes less than 1 hour to complete. The rendering is expected to be completed within a month. You need to minimize rendering costs.

What should you do?

a) Deploy the application as a managed instance group with Preemptible VMs.
b) Deploy the application as a managed instance group. Configure a Committed Use Discount for the amount of CPU and memory required.
c) Deploy the application as a managed instance group.
d) Deploy the application as a managed instance group with Preemptible VMs. Configure a Committed Use Discount for the amount of CPU and memory required.

Answer: a

Question: 5

You are running a production application on Compute Engine. You want to monitor the key metrics of CPU, Memory, and Disk I/O time.

You want to ensure that the metrics are visible by the team and will be explorable if an issue occurs. What should you do? (Choose 2)

a) Set up logs-based metrics based on your application logs to identify errors.
b) Export key metrics to a Google Cloud Function and then analyze them for outliers.
c) Set up alerts in Cloud Monitoring for key metrics breaching defined thresholds.
d) Create a Dashboard with key metrics and indicators that can be viewed by the team.
e) Export key metrics to BigQuery and then run hourly queries on the metrics to identify outliers.

Answer: c, d

Question: 6

Your application runs in Google Kubernetes Engine (GKE). You want to use Spinnaker with the Kubernetes Provider to perform blue/green deployments and control which version of the application receives traffic. What should you do?

a) Use a Kubernetes Replica Set and use Spinnaker to create a new service for each new version of the application to be deployed.
b) Use a Kubernetes Replica Set and use Spinnaker to update the Replica Set for each new version of the application to be deployed.
c) Use a Kubernetes Deployment and use Spinnaker to update the deployment for each new version of the application to be deployed.
d) Use a Kubernetes Deployment and use Spinnaker to create a new deployment object for each new version of the application to be deployed.

Answer: b

Question: 7

You support a Python application running in production on Compute Engine. You want to debug some of the application code by inspecting the value of a specific variable. What should you do?

a) Create a Cloud Debugger logpoint with the variable at a specific line location in your application's source code, and view the value in the Logs Viewer.
b) Use your local development environment and code editor to set up a breakpoint in the source code, run the application locally, and then inspect the value of the variable.
c) Modify the source code of the application to log the value of the variable, deploy to the development environment, and then run the application to capture the value in Cloud Logging.
d) Create a Cloud Debugger snapshot at a specific line location in your application's source code, and view the value of the variable in the Google Cloud Console.

Answer: d

Question: 8

Your Site Reliability Engineering team does toil work to archive unused data in tables within your application’s relational database. This toil is required to ensure that your application has a low Latency Service Level Indicator (SLI) to meet your Service Level Objective (SLO).

Toil is preventing your team from focusing on a high-priority engineering project that will improve the Availability SLI of your application.

You want to: (1) reduce repetitive tasks to avoid burnout, (2) improve organizational efficiency, and (3) follow the Site Reliability Engineering recommended practices.

What should you do?

a) Identify repetitive tasks that contribute to toil and onboard additional team members for support.
b) Identify repetitive tasks that contribute to toil and automate them.
c) Change the SLO of your Latency SLI to accommodate toil being done less often. Use this capacity to work on the Availability SLI engineering project.
d) Assign the Availability SLI engineering project to the Software Engineering team.

Answer: b

Question: 9

You support a website with a global audience. The website has a frontend web service and a backend database service that runs on different clusters. All clusters are scaled to handle at least ⅓ of the total user traffic.

You use 4 different regions in Google Cloud and Cloud Load Balancing to direct traffic to a region closer to the user.

You are applying a critical security patch to the backend database. You successfully patch the database in the first 2 regions, but you make a configuration error while patching Region 3. The unsuccessful patching causes 50% of user requests to Region 3 to time out.

You want to mitigate the impact of unsuccessful patching on users. What should you do?

a) Add more capacity to the frontend of Region 3.
b) Revert the Region 3 backend database and run it without the patch.
c) Drain the requests to Region 3 and redirect new requests to other regions.
d) Back up the database in the backend of Region 3 and restart the database.

Answer: c

Question: 10

You have an application deployed on Google Kubernetes Engine (GKE). The application logs are captured by Cloud Logging. You need to remove sensitive data before it reaches the Cloud Logging API.

What should you do?

a) Customize the GKE clusters’ Fluentd configuration with a filter rule. Update the Fluentd Config Map and Daemon Set in the GKE cluster.
b) Write the log information to the container file system. Execute a second process inside the container that will filter the sensitive information before writing to Standard Output.
c) Configure a filter in the Cloud Logging UI to exclude the logs with sensitive data.
d) Configure BigQuery as a sink for the logs from Cloud Logging, and then create a Data Loss Prevention job.

Answer: a

Avail the Study Guide to Pass Google GCP-PCDE Professional Cloud DevOps Engineer Exam:

- Find out about the GCP-PCDE syllabus topics. Visiting the official site offers an idea about the exam structure and other important study resources. Going through the syllabus topics help to plan the exam in an organized manner.
- Once you are done exploring the **GCP-PCDE syllabus**, it is time to plan for studying and covering the syllabus topics from the core. Chalk out the best plan for yourself to cover each part of the syllabus in a hassle-free manner.
- A study schedule helps you to stay calm throughout your exam preparation. It should contain your materials and thoughts like study hours, number of topics for daily studying mentioned on it. The best bet to clear the exam is to follow your schedule rigorously.
- The candidate should not miss out on the scope to learn from the GCP-PCDE training. Joining the Google provided training for GCP-PCDE exam helps a candidate to strengthen his practical knowledge base from the certification.
- Learning about the probable questions and gaining knowledge regarding the exam structure helps a lot. Go through the **GCP-PCDE sample questions** and boost your knowledge
- Make yourself a pro through online practicing the syllabus topics. GCP-PCDE practice tests would guide you on your strengths and weaknesses regarding the syllabus topics. Through rigorous practicing, you can improve the weaker sections too. Learn well about time management during exam and become confident gradually with practice tests.

Career Benefits:

- Passing the GCP-PCDE exam, helps a candidate to prosper highly in his career. Having the certification on the resume adds to the candidate's benefit and helps to get the best opportunities.