

Summary

Exit

Question 1:

You have an application that will run on Compute Engine. You need to design an architecture that takes into account a disaster recovery plan that requires your application to fail over to another region in case of a regional outage. What should you do?

- A. Deploy the application on two Compute Engine instances in the same project but in a different region. Use the first instance to serve traffic, and use the HTTP load balancing service to fail over to the standby instance in case of a disaster.
- B. Deploy the application on a Compute Engine instance. Use the instance to serve traffic, and use the HTTP load balancing service to fail over to an instance on your premises in case of a disaster.
- C. Deploy the application on two Compute Engine instance groups, each in the same project but in a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance group in case of a disaster.
- D. Deploy the application on two Compute Engine instance groups, each in separate project and a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance in case of a disaster.

Type : SINGLE SELECTION

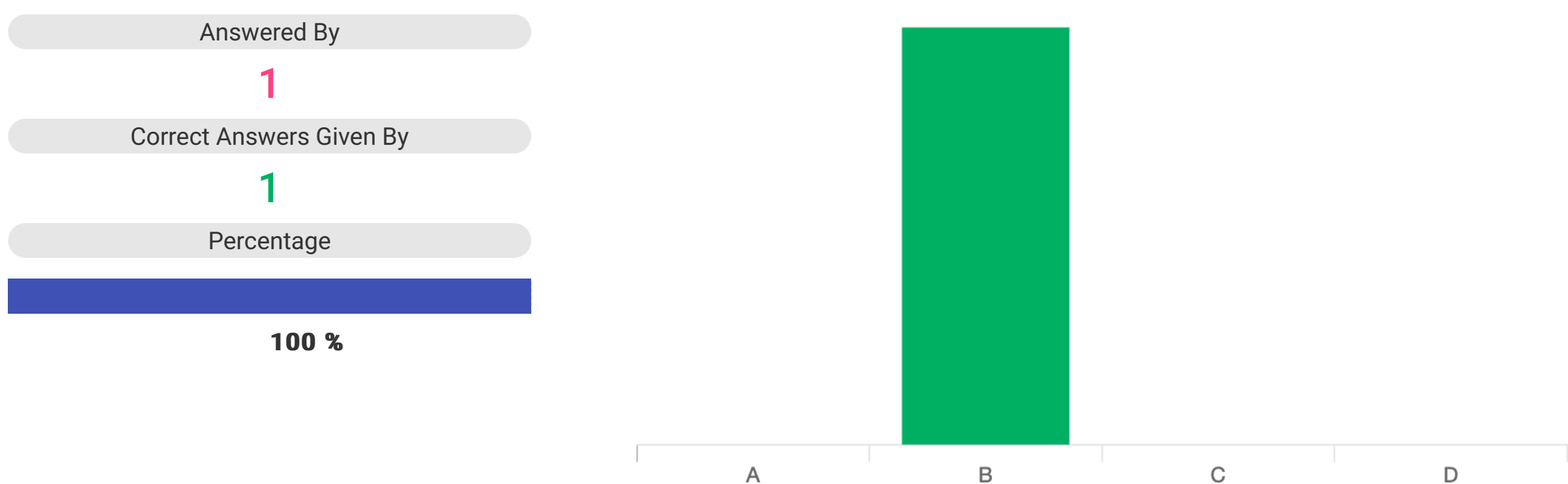


Question 2:

Your company's test suite is a custom C++ application that runs tests throughout each day on Linux virtual machines. The full test suite takes several hours to complete, running on a limited number of on-premises servers reserved for testing. Your company wants to move the testing infrastructure to the cloud, to reduce the amount of time it takes to fully test a change to the system, while changing the tests as little as possible. Which cloud infrastructure should you recommend?

- A. Google Compute Engine unmanaged instance groups and Network Load Balancer
- B. Google Compute Engine managed instance groups with auto-scaling
- C. Google Cloud Dataproc to run Apache Hadoop jobs to process each test
- D. Google App Engine with Google StackDriver for logging

Type : SINGLE SELECTION

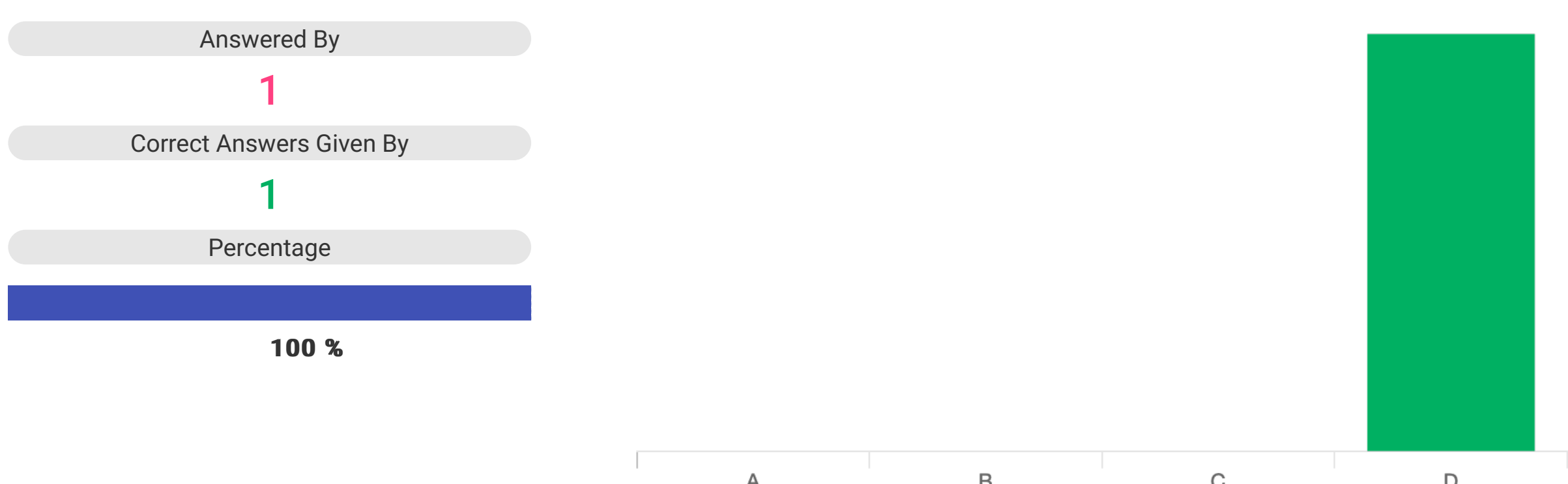


Question 3:

Your company has decided to make a major revision of their API in order to create better experiences for their developers. They need to keep the old version of the API available and deployable, while allowing new customers and testers to try out the new API. They want to keep the same SSL and DNS records in place to serve both APIs. What should they do?

- A. Configure a new load balancer for the new version of the API
- B. Reconfigure old clients to use a new endpoint for the new API
- C. Have the old API forward traffic to the new API based on the path
- D. Use separate backend pools for each API path behind the load balancer

Type : SINGLE SELECTION

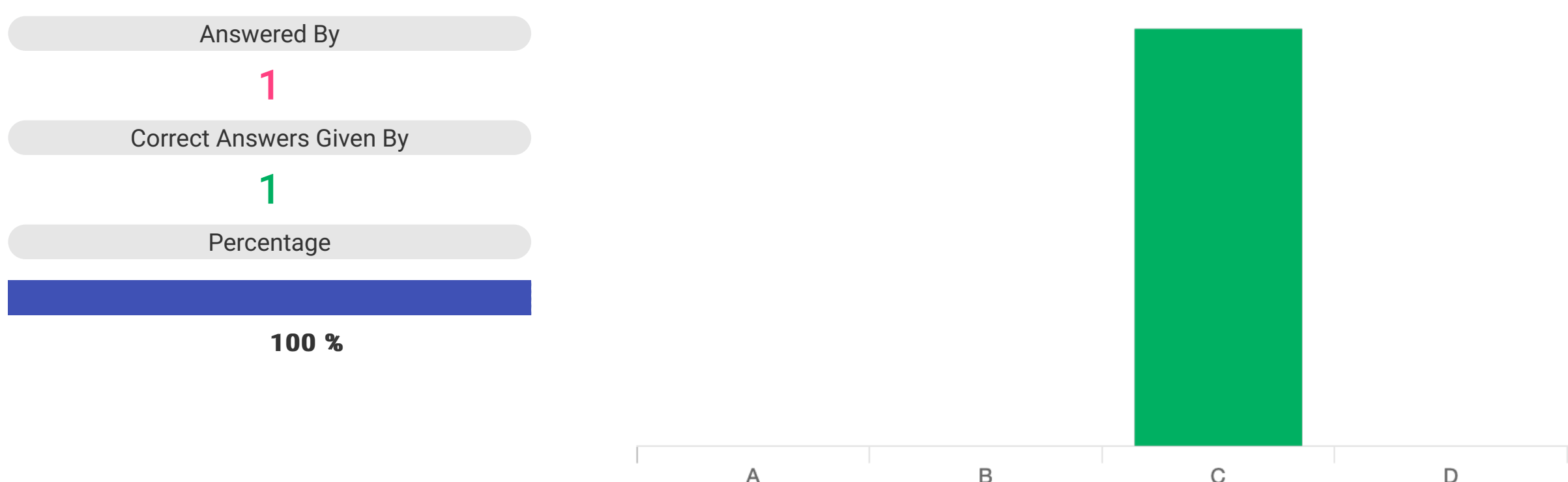


Question 4:

You set up an autoscaling managed instance group to serve web traffic for an upcoming launch. After configuring the instance group as a backend service to an HTTP(S) load balancer, you notice that virtual machine (VM) instances are being terminated and re-launched every minute. The instances do not have a public IP address. You have verified that the appropriate web response is coming from each instance using the curl command. You want to ensure that the backend is configured correctly. What should you do?

- A. Ensure that a firewall rule exists to allow source traffic on HTTP/HTTPS to reach the load balancer.
- B. Assign a public IP to each instance, and configure a firewall rule to allow the load balancer to reach the instance public IP.
- C. Ensure that a firewall rule exists to allow load balancer health checks to reach the instances in the instance group.
- D. Create a tag on each instance with the name of the load balancer. Configure a firewall rule with the name of the load balancer as the source and the instance tag as the destination.

Type : SINGLE SELECTION



Question 5:

Your company has a web application with many users all over the world. There are also huge spikes in demand at unpredictable times. High availability and low latency are extremely important as is cost. What would you recommend?

- A. Deploy the application to every data center. Then use Cloud DNS rules to route traffic to the closest server based on where the request came from.
- B. Set up the HTTP load balancer to balance traffic across instance groups in multiple regions around the world. Make sure there are enough instances running in each instance group to handle 120% of the peak load.
- C. Deploy your application to both AWS and Google Cloud data centers around the globe. Use the HTTP load balancer and set up global forwarding rules to send requests to each data center using a round-robin approach. Set up session affinity to ensure requests always go to the same server after the first request.
- D. Set up the HTTP load balancer to balance traffic across instance groups in multiple regions around the world. Set up autoscaling in the instance groups to turn machines on or off depending on load.

Type : SINGLE SELECTION

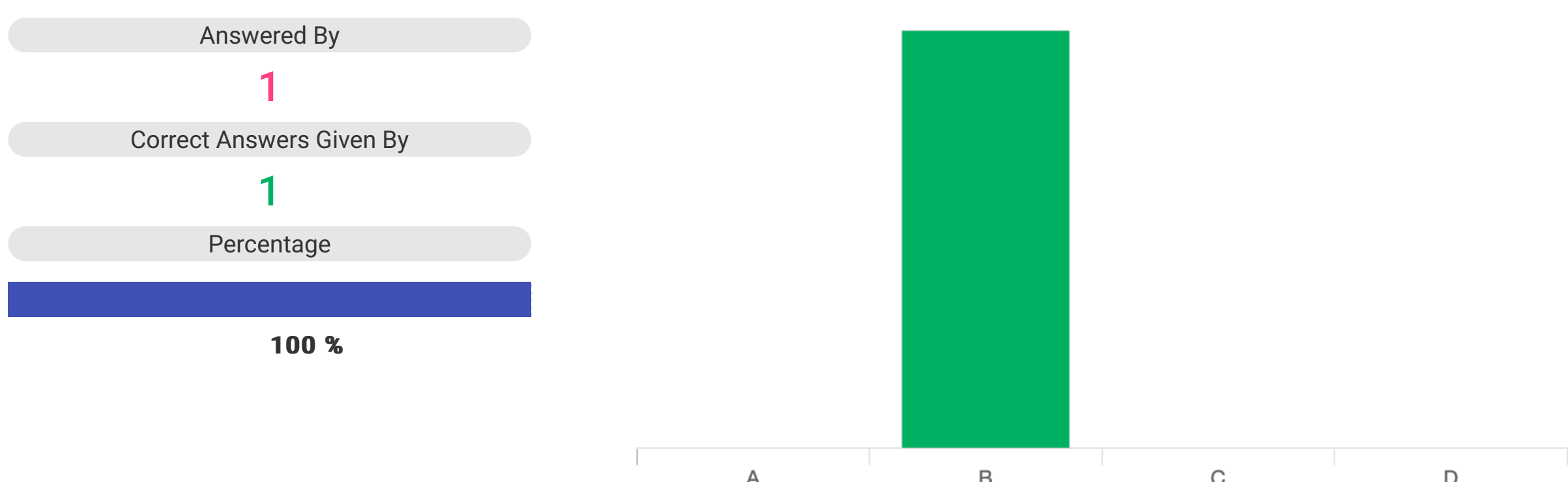


Question 6:

You need to design a solution for global load balancing based on the URL path being requested. You need to ensure operations reliability and end-to-end in- transit encryption based on Google best practices. What should you do?

- A. Create a cross-region load balancer with URL Maps.
- B. Create an HTTPS load balancer with URL maps.
- C. Create appropriate instance groups and instances. Configure SSL proxy load balancing.
- D. Create a global forwarding rule. Configure SSL proxy balancing.

Type : SINGLE SELECTION



Question 7:

You have an application that will run on Compute Engine. You need to design an architecture that takes into account a disaster recovery plan that requires your application to fail over to another region in case of a regional outage. What should you do?

- A. Deploy the application on two Compute Engine instances in the same project but in a different region. Use the first instance to serve traffic, and use the HTTP load balancing service to fail over to the standby instance in case of a disaster.
- B. Deploy the application on a Compute Engine instance. Use the instance to serve traffic, and use the HTTP load balancing service to fail over to an instance on your premises in case of a disaster.
- C. Deploy the application on two Compute Engine instance groups, each in the same project but in a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance group in case of a disaster.
- D. Deploy the application on two Compute Engine instance groups, each in separate project and a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance in case of a disaster.

Type : SINGLE SELECTION



Question 8:

Your company is running a stateless application on a Compute Engine instance. The application is used heavily during regular business hours and lightly outside of business hours. Users are reporting that the application is slow during peak hours. You need to optimize the application's performance. What should you do?

- A. Create a snapshot of the existing disk. Create an instance template from the snapshot. Create an autoscaled managed instance group from the instance template.
- B. Create a snapshot of the existing disk. Create a custom image from the snapshot. Create an autoscaled managed instance group from the custom image.
- C. Create a custom image from the existing disk. Create an instance template from the custom image. Create an autoscaled managed instance group from the instance template.
- D. Create an instance template from the existing disk. Create a custom image from the instance template. Create an autoscaled managed instance group from the custom image.

Type : SINGLE SELECTION

