DUMPSBOSS.

Salesforce Certified Heroku Architect

Salesforce Heroku-Architect

Version Demo

Total Demo Questions: 36

Total Premium Questions: 366

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Heroku Platform Architecture	97
Topic 2, Heroku Data	47
Topic 3, Heroku Security	40
Topic 4, Heroku Enterprise	43
Topic 5, Heroku Integrations	78
Topic 6, Mix Questions	61
Total	366

QUESTION NO: 1

A client wants to deconstruct a monolithic app into a collection of smaller apps which should be able to send HTTP requests to each other. The smaller apps should not be publicly accessible on the internet. Which Heroku Enterprise features should an Architect recommend to enable this architecture?

A. Heroku and ApacheKafka on Heroku

B. Heroku Connect and Internal Routing

C. Heroku Private Spaces and Heroku Connect

D. Heroku Private Spaces and Internal Routing

ANSWER: D

Explanation:

Heroku Private Spaces and Internal Routing is the appropriate combination for this microservices-style architecture. A Private Space provides an isolated Heroku runtime environment with its own network, allowing an organization to deploy applications that are not exposed through the public internet by default. This is the required enterprise foundation when the services must remain private. Internal Routing then enables applications in the same Private Space, or in spaces connected through the appropriate network configuration, to make HTTP requests to one another using internal DNS hostnames. Requests remain on Heroku's private network rather than traversing the public internet. Each service can therefore expose an internal web process and be addressed by its internal application name, supporting direct service-to-service HTTP communication while preserving the intended network boundary. This design lets the client split the monolith into independently deployed apps without making each backend service externally reachable. Access controls, private-space network isolation, and internal service discovery collectively support a secure decomposition strategy. See Heroku's Private Spaces documentation and its guide to Private Space Internal Routing.

QUESTION NO: 2

A client runs an application on a background worker dyno. The application allows its users to request personalized information as a PDF. At peak usage, the app processes millions of requests at once. The resulting number of requests has caused a bottleneck that is impacting its performance.

Which two solutions can an Architect recommend to resolve the bottleneck and improve performance? (Choose two.)

A. Increase the number of workers to consume the job faster.

B. Move the application to a Private Space.

C. Add Heroku Redis as a job queue.

D. Add a CDN add-on from the Elements marketplace.

ANSWER: A C

Explanation:

Increase the number of workers to consume the job faster. is appropriate because worker dynos are intended for asynchronous, long-running, or resource-intensive work outside the web request path. Scaling the worker process horizontally increases the number of jobs that can be processed concurrently, raising throughput and reducing the time jobs remain backlogged during peak demand. The worker count should be selected with consideration for downstream limits, such as database capacity and external PDF-generation services.

Add Heroku Redis as a job queue. is also appropriate because a durable asynchronous queue separates the burst of PDF requests from their execution. The application can enqueue each personalized-PDF job quickly, while worker dynos consume jobs at a controlled, scalable rate. Heroku Data for Redis is commonly used as the backing store or broker for job-processing frameworks, enabling the architecture to absorb very large traffic spikes without requiring every request to be processed immediately. Together, queue-based buffering and horizontally scaled workers provide a resilient producer-consumer design for this workload. See Heroku Background Jobs and Queueing and Heroku Data for Redis.

QUESTION NO: 3

A company currently uses, and will continue to use, the MulesoftAnypoint Platform integration throughout their architecture. They want to use Heroku Connect to help them integrate Heroku apps with Salesforce apps. What is a benefit of introducing Heroku Connect in this environment?

A. Heroku Connect provides Salesforce a bidirectional synchronization between Platform Events on Salesforce and Apache Kafka on Heroku

B. HerokuConnect provides Mulesoft an endpoint that is easily consumable by a back-end integration tool

C. Heroku Connect provides Salesforce a declarative bidirectional synchronization with Postgres and data proxy capability via Heroku Connect External Objects.

D. Heroku Connect has a Microsoft connector for access to Heroku Postgres Heroku Redis, and Apache Kafka on Heroku

ANSWER: C

Explanation:

Heroku Connect provides Salesforce a declarative bidirectional synchronization with Postgres and data proxy capability via Heroku Connect External Objects. This is valuable because it gives the organization a purpose-built data-integration layer between Salesforce objects and a Heroku Postgres database while allowing MuleSoft Anypoint Platform to remain in place for the broader enterprise integration architecture. Administrators configure mappings declaratively, selecting the Salesforce objects and fields to synchronize with Postgres tables, rather than requiring application code to continually move operational data between the two platforms. Heroku applications can therefore work with relational Postgres data locally and efficiently, while Salesforce users and processes continue to work with the corresponding CRM records.

Heroku Connect supports synchronization in either direction, according to the mapping configuration, and handles routine data replication between the Salesforce organization and the Heroku Postgres database. Its External Objects capability further supports a data-proxy pattern, enabling Salesforce to expose externally stored data without copying all of that data into standard Salesforce objects. This combination is particularly appropriate when an architecture needs both Salesforce-native access to relevant application data and scalable Heroku-based processing. See the Heroku Connect Dev Center documentation and Salesforce's Salesforce Connect overview.

QUESTION NO: 4

Universal Containers intends to build an app which will accept card payments. The app also needsto store, process, and transmit cardholder data.

Which Heroku architecture should an Architect recommend?

A. Common Runtime with secure, isolated containers for running theapp's code.

B. A Private Space restricted to a set of trusted IP ranges.

C. A Shield Private Space with a Shield Postgres add-on.

D. A Private Space with Internal Routing enabled on the app.

ANSWER: C

Explanation:

A Shield Private Space with a Shield Postgres add-on is the appropriate architecture because the application handles cardholder data throughout the payment lifecycle: accepting, storing, processing, and transmitting it. That scope requires a platform architecture designed to support Payment Card Industry Data Security Standard obligations and to provide stronger controls around the application and its data services. Heroku Shield Private Spaces are Heroku's compliance-focused isolated runtime environment and are intended for workloads subject to stringent regulatory requirements, including PCI. A Shield Private Space provides the required protected application boundary, while Shield Postgres extends the Shield environment to the database layer where cardholder data is stored and processed. Using both services keeps the application and its persistent data within the Shield product boundary, rather than combining a compliant runtime with a non-Shield data

store. The architecture should still be paired with the customer's own PCI controls, such as secure application development, access management, encryption practices, logging, vulnerability management, and properly scoped payment integrations. Heroku documents the compliance capabilities and responsibilities of Shield services at Heroku Shield and identifies Shield Private Spaces and Shield Postgres as Shield products in its security documentation.

QUESTION NO: 5

Universal Containers wants to reduce their mean-time-to-service

Which three Field Service process should a Consultant recommend to accomplish this goal? (Choose three)

A. Knowledge Base

B. Customer Entitlements

C. Adjust Scheduling Policy

D. Dispatching

ANSWER: A C D

Explanation:

Reducing mean time to service requires getting the right resource to the job quickly and enabling that resource to complete the work efficiently on the first visit. **Adjust Scheduling Policy** supports this objective by defining the criteria used to rank service appointments, such as priority, due date, travel time, skills, operating hours, and service territory. A well-designed policy helps optimization and scheduling select appointments that can be completed sooner while respecting operational constraints.

Dispatching is also essential because dispatchers can monitor the schedule, respond to urgent changes, assign or reassign appointments, and use the scheduling tools to minimize delays. This operational control helps keep work moving when cancellations, emergencies, or resource availability changes occur.

Knowledge Base improves technician efficiency by making approved troubleshooting, diagnostic, and repair guidance available during service work. Faster access to consistent instructions reduces time spent investigating issues and increases the likelihood of resolving work without additional visits. Together, scheduling policy, active dispatching, and accessible knowledge address the major contributors to service duration: assignment speed, travel and scheduling efficiency, and time spent completing the repair. See Salesforce's Field Service scheduling and optimization documentation and Knowledge documentation.

QUESTION NO: 6

Which two statements describe characteristics of Private Space Logging? (Choose two.)

A. Log entries are forwarded as HTTPS POST requests.

B. Log entries are collated and forwarded by Heroku's Logplex router.

C. Log entries can be forwarded to up to three logging services.

D. Log entries can be forwarded to only one logging service.

ANSWER: A C

Explanation:

Private Space Logging supports sending a Space's operational log stream to external logging endpoints through HTTPS log drains. Each log entry is delivered as an HTTPS POST request, enabling customers to receive and process logs in a centralized observability, security-monitoring, or archival platform. This delivery model is particularly useful for Private Spaces because it provides an outbound, encrypted integration path to a customer-managed logging service without requiring access to individual application dynos.

A Private Space can have as many as three logging endpoints configured. This permits the same Space-level stream to be delivered to multiple destinations, such as a primary log-management platform, a security information and event management system, and a separate retention or analytics service. The endpoints are configured at the Space level, so the configured logging integration applies to apps running within that Private Space. Heroku documents the HTTPS delivery mechanism and the maximum of three logging endpoints in its Private Space Logging documentation. See Heroku Private Space Logging and Heroku Log Drains.

QUESTION NO: 7

Universal Containers wants their Field Technicians to indicate if any of their Service Appointments are at risk of not being completed on time. They would like forthis to be achieved on a mobile device using a Quick Action

What should a Consultant recommend to achieve this requirement?

A. Update the Service Appointment Status field.

B. Update the Service Appointment Chatter feed.

C. Reschedule the Service Appointment for later

D. Update the Service Appointment field "In Jeopardy"

ANSWER: D

Explanation:

Updating the Service Appointment field "In Jeopardy" directly captures the required operational signal: that the appointment is at risk of not being completed within its expected timeframe. In Field Service, the Service Appointment record represents the scheduled work that technicians perform, so placing the risk indicator on that record makes the information immediately available to dispatchers, supervisors, automation, and reporting. A consultant can expose this checkbox in a mobile record-update quick action so a technician can flag the appointment with minimal effort while working in the field. The quick action can be added to the relevant Service Appointment mobile action layout, allowing the technician to update the current appointment without navigating through a full edit experience. Because the risk state is stored as structured record data, the organization can also use it in list views, reports, flows, notifications, and dispatch processes to identify work requiring attention. Salesforce's Field Service documentation describes Service Appointments as the records used to schedule and track service work; see ServiceAppointment object reference. Salesforce also documents how quick actions provide focused record-update interactions in mobile apps: Create Quick Actions.

QUESTION NO: 8

Universal Containers has a Heroku app that uses Heroku Connect to sync data with their Salesforce org. The app makes frequemt updates to the same records over short period of time, and sync speeds have begun to worsen. An Architect recommends using Heroku Connect's Merged Writes algorithm to improve sync speeds. What are 2 implications of using the Merged Writes algorithm in this scenario?

A. Relationships such ascircular dependencies are known to cause issues.

B. The Merged Writes algorithm does not support using Salesforce SOAP API

C. The Merged Writes algorithm ismore likely to approach Salesforce API rate limits.

D. The Merged Writes algorithm does not supportusing the Salesforce Bulk API.

ANSWER: A D

Explanation:

"Relationships such ascircular dependencies are known to cause issues." and "The Merged Writes algorithm does not supportusing the Salesforce Bulk API." are correct. Merged Writes is designed for workloads that repeatedly modify the same mapped record over a short interval. Rather than sending every individual database change to Salesforce, Heroku

Connect combines pending changes for a record into a consolidated write. This reduces redundant outbound updates and can improve throughput for a high-churn application.

Because the algorithm must coordinate and combine writes while maintaining relationship ordering, complex relationship structures need careful design. In particular, circular dependencies can make it difficult to determine a valid sequence for applying related record changes, which can cause synchronization issues. Teams using this algorithm should review mapped object relationships and avoid designs that require mutually dependent records to be written in an unresolved order.

Merged Writes also uses Salesforce's SOAP-based write path rather than the Salesforce Bulk API. Consequently, it cannot be combined with Bulk API write behavior. This implementation detail matters when selecting a write algorithm: Merged Writes is optimized for coalescing frequent updates to the same records, not for bulk-loading large, independent batches. See Heroku Connect Write Algorithms and Heroku Connect documentation.

QUESTION NO: 9

Universal Containers is developing a Salesforce app thatinvokes a Heroku app ' s web service, which asynchronously generates customer invoices. The Heroku app is deployed to a Private Space. When an invoice is ready, the Heroku app sends a POST request to the Salesforce REST API. Which two options should an Architect recommend to ensure that neither the Salesforce nor the Heroku app is accessible from the public internet? Choose 2 answers.

A. Restrict the Private Space ' s trusted IP range to Salesforce IP addresses

B. Restrict the Private Space ' s trusted IP range to Universal Containers ' VPN

C. Restrict the Salesforce connected app ' s login IP ranges to Universal Containers ' VPN

D. Restrict the Salesforce connected app ' s login IP ranges to the stable outbound IP addresses of the Private Space

ANSWER: A D

Explanation:

Restricting the Private Space's trusted IP range to Salesforce IP addresses establishes an inbound allowlist for the Heroku application. A Private Space provides stable networking controls, and its trusted IP ranges limit which source IP addresses can reach applications in the space. Allowlisting Salesforce's published outbound IP ranges permits the Salesforce-originated request to reach the invoice service while preventing arbitrary public sources from accessing it.

Restricting the Salesforce connected app's login IP ranges to the stable outbound IP addresses of the Private Space provides the reciprocal control for the asynchronous callback. Private Spaces provide stable outbound IP addresses (NAT addresses), so Salesforce can reliably identify calls made from the Heroku app. Configuring those addresses on the connected app means the OAuth/API access used for the REST POST is accepted only when it originates from the Private Space's controlled egress addresses. Together, these controls create a mutually restricted integration path: Salesforce can call the Heroku service, and the Heroku service can call Salesforce, without opening either side to unrestricted internet clients. Heroku documents this exact trust pattern at Establish Trust Between a Private Space and Salesforce. See also Private Spaces stable outbound IP addresses.

QUESTION NO: 10

A client provisions a Heroku Postgres database in the EU region.

Which two services related to Heroku Postgres are located in the U.S.? (Choose two.)

A. Fork and follower databases

B. Heroku Postgres Continuous Protection backups

C. Snapshots created with Heroku PGBackups

D. Heroku Dataclips

ANSWER: A D

Explanation:

For an EU-region Heroku Postgres database, **Fork and follower databases** and **Heroku Dataclips** are services whose data is located in the United States. This is an important data-residency consideration: although the provisioned database is in the EU, creating a fork or follower uses infrastructure located in the U.S. A fork is an independent copy of a database, while a follower is a continuously updated replica, so either service can create a U.S.-located copy of data from the EU database.

Heroku Dataclips also operate from U.S.-based infrastructure. A Dataclip saves a SQL query and exposes its result through a shareable URL, meaning that its use can involve database result data in the U.S. Organizations with residency, privacy, or regulatory obligations must account for these service-specific locations rather than relying solely on the region of the primary Heroku Postgres database. Salesforce documents these regional considerations for Heroku Postgres and describes Dataclips and their query-result behavior in the Heroku Dev Center. See Heroku Postgres Data Residency and Heroku Dataclips.

QUESTION NO: 11

A Heroku application uses Heroku Connect to integrate with data from a Salesforce org. Since then, several fields from the Salesforce objects that were mapped using Connect have been removed. As a result, the development team will reload the mapping using Heroku Connect.

What is an implication of reloading the mapping?

A. The mapped tables in Heroku Postgres will continue to store the previously synchronized data, and synchronization will restart normally.

B. The mapped tables in Heroku Postgres will be truncated as part of the reload action. The data in removed fields will be lost, and synchronization will pause until manually restarted.

C. The mapped tables in Heroku Postgres will continue to store the previously synchronized data, but synchronization will pause until manually restarted.

D. The mapped tables in Heroku Postgres will be truncated as part of the reload action. The data in removed fields will be lost, and synchronization will restart normally.

ANSWER: D

Explanation:

Reloading a Heroku Connect mapping truncates the associated mapped table in Heroku Postgres and then repopulates it from Salesforce. This is important when mapped Salesforce fields have been removed: the reload rebuilds the local representation using the current mapping and source-object structure, so values that existed only in the removed PostgreSQL columns are no longer retained. A reload is therefore a destructive operation for the mapped table's existing contents; applications should not rely on the table as the sole copy of any data that cannot be obtained again from Salesforce.

After the reload is initiated, Heroku Connect performs the initial synchronization again and resumes normal ongoing synchronization automatically when that process completes. The mapping does not require a separate manual restart merely because it was reloaded. Teams should plan for the reload window, particularly for large objects, because the mapped table is emptied before Salesforce records are loaded back into Postgres. Heroku's documentation describes reload as reloading all records for a mapping, while the mapping documentation covers how mapped Salesforce objects are represented and synchronized in the Heroku Postgres database. See Heroku Connect Mappings and Heroku Connect.

QUESTION NO: 12

The Private Spaces feature can be useful if you need to:

A. None of these

B. Write Apex applications quickly

C. Speed up an application's response time by running it on dynos that are located geographically closer to your customers

D. Ensure that your application's incoming traffic originates from a whitelisted set of IP addresses

ANSWER: C D

Explanation:

Private Spaces are useful for deploying an application in a selected Heroku region, which can reduce network latency for customers located nearer to that region. The application's dynos, routing, and supporting Private Space infrastructure are provisioned within the chosen geographic region. Selecting a region aligned with the application's primary users is therefore an important architecture decision when response time is sensitive to geographic distance.

Private Spaces also support trusted IP ranges. These ranges allow an organization to limit access to applications in the space so that requests must come from specified IP addresses or CIDR blocks. This is valuable when an application is intended for a corporate network, VPN, partner network, or other controlled client environment. Trusted IP ranges provide a network-level access-control layer at the Heroku router, helping ensure that publicly reachable application endpoints accept traffic only from approved source networks. The ranges can be managed for the space and applied consistently to its apps. See Heroku's Private Spaces documentation and its guidance on trusted IP ranges.

QUESTION NO: 13

A client is building a collection of web apps that will serve short-lived marketing sites during REST. The apps must support HTTPS connections. They also need to scale up and down at unpredictable intervals. The client is not currently a Heroku Enterprise customer. Given the scenario, what should an Architect recommend?

A. Sign up for Heroku Enterprise and deploy each app to a different Private Space

B. Sign up for Heroku Enterprise without deploying the apps to the same Private Space

C. Sign up for Heroku Enterprise and deploy the apps to the same Private Space

D. Deploy the apps to the Common Runtime

ANSWER: D

Explanation:

Deploy the apps to the Common Runtime. The Common Runtime is Heroku's multitenant application environment and is appropriate for independently deployed, internet-facing marketing applications that do not require the networking isolation and enterprise controls of a Private Space. Each application can use Heroku's routing layer and be configured for secure HTTPS traffic by adding a verified custom domain and using Heroku SSL or Automated Certificate Management. This supports the essential requirement to serve sites securely without requiring an Enterprise account.

The applications can also be scaled independently as traffic changes. Heroku dynos are the basic unit of compute, and their quantity can be increased or decreased for each app without changing the application architecture. For unpredictable demand, the client can use Heroku's autoscaling capabilities where available for the selected dyno type, or automate dyno formation changes through the Platform API. This makes the Common Runtime a practical fit for short-lived campaign sites: applications remain isolated at the app level, can be created and removed as campaigns begin and end, and can adjust compute capacity according to demand. See Heroku SSL documentation and Heroku dyno scaling documentation.

QUESTION NO: 14

Which three items are required to successfully set up Single Sign-on (SSO)services with Heroku? (Choose three.)

A. An identity provider that supports SAML 2.0

B. A Heroku Enterprise Team

C. An existing Heroku account for each user

D. At leastone valid SSO certificate

E. Administrative permissions on the selectedidentity provider

ANSWER: A B E

Explanation:

Successfully configuring Heroku single sign-on requires a compatible identity provider, a Heroku Enterprise Team, and administrative permissions in the selected identity provider. The identity provider must support the SAML 2.0 standard used by Heroku for federation. This capability allows the provider to authenticate a user and send the required signed SAML assertions to Heroku, so users can access Heroku through the organization's centralized authentication process.

A Heroku Enterprise Team is required because SSO configuration is an enterprise feature managed at the team level. The team is the Heroku resource to which the organization's SSO settings and membership controls apply. Administrative permissions on the identity provider are also necessary because the administrator must create or configure the Heroku application integration, establish the SAML connection details, and manage the provider-side settings required for the trust relationship.

Heroku's setup guidance identifies these prerequisites and provides the configuration process for SSO services. See Using SSO Services with Heroku and Heroku Enterprise Teams.

QUESTION NO: 15

Universal Containers needs to integrate three separate apps: a Salesforce app, an app developed to Heroku, and a custom-built app hosted on-premise. Each app needs to send and receive data from any of the other apps. Which two integration options should an Architect recommend that can connect all three systems in a fire-and-forget pattern? Choose 2 answers.

A. Publish and subscribe to Salesforce platform events

B. Use Heroku External Objects as a data proxy

C. Use Heroku Connect to synchronize all systems with Heroku Postgres

D. Publish and subscribe to topics on Apache Kafka on Heroku

ANSWER: A D

Explanation:

Publish and subscribe to Salesforce platform events provides an event-driven, asynchronous integration mechanism. Salesforce applications can publish events, while external clients—including applications running on Heroku and on-premises services—can publish to and subscribe for events through Salesforce APIs. Producers do not wait for consumers to complete processing, so this supports the required fire-and-forget communication model while allowing each system to react independently to events. Salesforce documents platform events as part of its event-driven architecture and supports external event publishers and subscribers through the Pub/Sub API and streaming interfaces. See Salesforce Pub/Sub API Overview.

Publish and subscribe to topics on Apache Kafka on Heroku also supplies a durable, decoupled publish/subscribe event backbone. The Salesforce application, Heroku application, and on-premises application can use appropriate producers, consumers, and integration connectors to exchange messages through Kafka topics. A producer writes an event to a topic without requiring an immediate response from downstream consumers; consumers process events independently and can scale separately. This makes Kafka well suited to bidirectional, many-to-many integration across these three environments. Kafka's architecture is based on publishers writing records to topics and subscribers reading them independently, as described in the Apache Kafka documentation.

QUESTION NO: 16

A client's Heroku web application displaysdata that is fetched from a back-end file storage system. The client now wants this data to be viewable, but not editable, from their Salesforce org.

Which recommendation should an Architect make in this scenario?

A. Replicate the files to tables in a Heroku Postgres database, and use Heroku Connect to synchronize the tables to the Salesforce org.

B. Store all file URLs in a Heroku Postgres table, and use Heroku Connect to synchronize the table to the Salesforce org.

C. Replicate the files to tables in aHeroku Postgres database, and use Heroku External Objects to expose the tables to the Salesforce org.

D. Replicate the files to the application's local filesystem, and use worker dynos to periodically sync them to the Salesforce org.

ANSWER: C

Explanation:

Replicate the files to tables in aHeroku Postgres database, and use Heroku External Objects to expose the tables to the Salesforce org. This approach is appropriate because Heroku External Objects makes data held in Heroku Postgres available in Salesforce as external objects. Salesforce users can view the externally stored data without making Salesforce the system of record or creating a synchronized copy of the data in Salesforce. The data remains managed by the Heroku application and its backing storage design, while Salesforce provides an integrated interface for users who need visibility.

External objects are designed for accessing external data on demand through Salesforce Connect. This is particularly suitable for a view-only requirement, because the integration can expose the required file metadata or content representation from the Heroku-managed data store while avoiding unnecessary bidirectional synchronization and associated data duplication. The architect should model the replicated file information in Postgres in a form that can be exposed as external-object records, including identifiers and the attributes Salesforce users must view. Heroku documents the Heroku External Objects capability at Heroku External Objects, and Salesforce describes the external-data access model in its External Objects documentation.

QUESTION NO: 17

A client wants to deconstruct a monolithic application into a set of microservices The microservices require securer direct peer-to-peer communications.

Which Heroku Enterprise features should an Architect recommend? Choose one answer

A. Heroku Private Spaces and Apache Kafka on Heroku

B. Shield Private Spaces and Heroku Shield Connect

C. Heroku Private Spaces and Internal Routing.

D. Heroku Private Spaces and Private Space VPN connections.

ANSWER: C

Explanation:

Heroku Private Spaces and Internal Routing is the appropriate combination for microservices that need secure, direct service-to-service communication. A Private Space provides an isolated, dedicated network environment for the client's Heroku applications. Within that space, Internal Routing lets applications communicate by using their internal `herokuapp.com` DNS names rather than sending requests through the public routing layer. This keeps traffic on the space's private network and avoids exposing those service endpoints to the public internet. It also supports a microservices architecture cleanly: each service can remain independently deployed as its own Heroku app while calling peer services through internal URLs. Internal Routing is enabled at the Private Space level, and applications in the same space can then use internal endpoints for private communication. This design gives the organization network isolation for its application workload along with straightforward east-west connectivity between the decomposed services. For implementation details and requirements, see Heroku Private Spaces: Internal Routing and Heroku Private Spaces.

QUESTION NO: 18

In the hierarchy of cloud services, PaaS generally provides more out of the box than:

A. Software as a service (SaaS)

B. Infrastructure as a service (IaaS) and on-premises infrastructure

C. Salesforce Marketing Cloud

D. Visualforce pages

ANSWER: B

Explanation:

Infrastructure as a service (IaaS) and on-premises infrastructure is correct because platform as a service (PaaS) supplies a managed application platform on top of the underlying computing infrastructure. Rather than requiring teams to assemble and operate core platform capabilities themselves, a PaaS typically delivers runtime environments, application deployment workflows, scaling mechanisms, routing, logging, and managed operational services as built-in platform features. This enables developers to concentrate primarily on application code and data rather than provisioning servers, configuring operating systems, patching platform software, and building deployment tooling.

Heroku is an example of PaaS: applications are deployed to the platform, which runs them in dynos and provides platform services such as routing, configuration through config vars, process management, and an ecosystem of add-ons. The National Institute of Standards and Technology similarly defines PaaS as a cloud model in which the consumer deploys applications using provider-supported languages, libraries, services, and tools while the provider manages the underlying cloud infrastructure. Accordingly, PaaS offers more ready-to-use application-platform functionality than infrastructure-focused and self-managed environments. See Heroku Dev Center reference documentation and NIST SP 800-145 cloud-computing definition.

QUESTION NO: 19

Which of the following is a Heroku CLI command?

A. heroku create

B. heroku new

C. git push heroku master

D. git push

ANSWER: A

Explanation:

`heroku create` is a Heroku CLI command that creates a new Heroku application. When run without an app name, it provisions an app with a generated name; when an available name is supplied, it creates the app using that name. The command also configures a Git remote named `heroku` for the newly created application when used in a Git repository, allowing subsequent deployments to target that Heroku app. This is part of the standard Heroku Command Line Interface workflow: install the CLI, authenticate with `heroku login`, create an application, and then manage or deploy it using Heroku tooling. The Heroku CLI documentation lists `heroku create` as the command for creating an app and documents its available parameters, such as specifying a region, stack, team, or app name. See the Heroku CLI documentation and the Heroku guide to creating apps.

QUESTION NO: 20

Universal Containers has a Heroku app that uses Heroku Connect to sync data with their Salesforce org. The app makes frequemt updates to the same records over short period of time, and sync speeds have begun to worsen. An Architect recommends using Heroku Connect's Merged Writes algorithm to improve sync speeds. What are 2 implications of using the Merged Writes algorithm in this scenario?

A. Relationships such as circular dependencies are known to cause issues.

B. The Merged Writes algorithm does not support using Salesforce SOAP API

C. The Merged Writes algorithm is more likely to approach Salesforce API rate limits.

D. The Merged Writes algorithm does not support using the Salesforce Bulk API.

ANSWER: A D

Explanation:

“Relationships such as circular dependencies are known to cause issues.” is correct because merged writes consolidate repeated changes to a mapped record before sending the resulting update to Salesforce. This improves efficiency for a workload that repeatedly updates the same rows in a short interval, but relationships require careful mapping and ordering. In particular, circular dependencies can prevent Heroku Connect from determining a safe sequence for writing related records, so architects must design and test mappings with those dependencies in mind.

“The Merged Writes algorithm does not support using the Salesforce Bulk API.” is also correct. Merged writes are processed through Salesforce’s SOAP API rather than the Bulk API. The SOAP-based behavior supports the algorithm’s purpose: coalescing multiple pending row changes into a smaller number of record updates. It is therefore suitable when the primary issue is frequent, repeated updates to the same records, while the integration design must account for the API behavior and relationship constraints of this write mode. Salesforce documents the available Heroku Connect write algorithms and their API characteristics in the Heroku Connect Write Algorithms guide. For broader configuration and mapping considerations, see the Heroku Connect documentation.

QUESTION NO: 21

The primary benefit of a workflow outbound message over an Apex trigger is:

A. The message supports different authentication mechanisms

B. The message can deliver the payload only as SOAP

C. Messages are completely declarative.

D. The message can handle every database event.

ANSWER: C

Explanation:

“Messages are completely declarative.” is correct because Salesforce outbound messaging is configured through point-and-click automation rather than custom Apex code. An administrator defines the outbound message, selects the object fields to include, and associates it with a workflow rule or other supported declarative automation. Salesforce then sends a SOAP notification containing the selected record data to the configured endpoint when the automation criteria are met.

This declarative approach is especially valuable when the integration requirement is straightforward: notify an external system when a record reaches a particular business state. It reduces implementation effort, avoids the need to write and maintain trigger code, and lets admins review or adjust the automation using standard Salesforce setup tools. Salesforce also manages outbound-message delivery behavior, including queued delivery and retries for unavailable endpoints, as part of the platform capability. The receiving service can acknowledge successful processing through the outbound messaging SOAP interface.

For the Salesforce implementation model and SOAP notification details, see the Salesforce Outbound Messaging documentation and the Salesforce Trailhead outbound messaging unit.

QUESTION NO: 22

A client has data in a Heroku Postgres table. They want to generate analytics based on the table and make the results available in their Salesforce org. Their Salesforce administratorwants to minimize the amount of data that is written to Salesforce to accomplish this. Which strategy should an Architect recommend on this scenario?

A. Use Heroku Connect to sync the Heroku Postgres table to the Salesforce org, and generate analytics fromthe Salesforce org.

B. Use the Heroku application to generate analytics, and write the results to a separate Heroku Postgres table-Expose that table to the Salesforce org with Heroku External Objects.

C. Use the Heroku application to generate analytics, and write the results to the Salesforce org with the Salesforce Bulk API.

D. Use the Heroku application to generate analytics and write them to a separate Heroku Postgres table. Sync that table to the Salesforce org with Heroku Connect.

ANSWER: B

Explanation:

Use the Heroku application to generate analytics, store the resulting data in a separate Heroku Postgres table, and expose that table to Salesforce with Heroku External Objects. This approach keeps both the source data and computed analytics in Heroku Postgres, where the application can efficiently run the required processing. Salesforce users can then access the analytics as external objects without copying the full result set into Salesforce custom objects. Because Salesforce accesses the external data through Salesforce Connect at query time, this strategy minimizes Salesforce data storage and avoids the ongoing writes that a synchronization or API-based import would require. External objects can be used in the Salesforce interface and can participate in supported relationships and reporting capabilities, while the authoritative analytics data remains in the external system. This design is especially appropriate when Salesforce needs visibility into Heroku-hosted results rather than local persistence of those results. See Salesforce's External Objects documentation and Heroku's Heroku Connect documentation for details on external access and synchronization patterns.

QUESTION NO: 23

A client wants to secure their web application using SSL/TLS. They do not have specific requirements regardingusing a particular version of SSL/TLS.

Which two Heroku features should an Architect recommend as options? (Choose two.)

A. Heroku Private Spaces

B. Automated Certificate Management

C. Heroku SSL

D. Heroku Shield Private Spaces

ANSWER: B C

Explanation:

Automated Certificate Management is an appropriate option when the application uses a custom domain and the client wants Heroku to obtain, install, and renew the TLS certificate automatically. ACM uses Let's Encrypt certificates and handles renewal without requiring the team to manually track certificate expiration. It is available for eligible applications using Heroku's Common Runtime or Private Spaces and provides a straightforward managed approach to HTTPS.

Heroku SSL is also appropriate when the client wants to secure traffic to a custom domain but needs to use a certificate obtained and managed outside of ACM. This feature enables the team to upload a certificate and private key, then associate the certificate with the app's SSL endpoint. It therefore supports organizations whose certificate procurement, validation, or renewal processes are managed externally. Both capabilities terminate TLS at Heroku's routing layer and provide encrypted HTTPS access for the application; the choice depends primarily on whether Heroku-managed certificates or customer-managed certificates are preferred. See Heroku's Automated Certificate Management documentation and its SSL Endpoint documentation.

QUESTION NO: 24

A client is writing a Heroku application that requirescompliance with PCI DSS Level 1. To accomplish this, they deploy an application to the ShieldPrivate Space. Which statement is true about the application?

A. The application might still violate PCI DSS Level 1 even though it is deployed to a Shield Private Space

B. The Shield Private Space guarantees that the application is compliant with PCI DSSLevel 1, assuming the application uses only Shield Postgres databases

C. The application is definitely not compliant with PCI DSS Level 1. because Shield PrivateSpaces do not provide compliance with it.

D. The Shield Private Space guarantees that the application is compliant with PCI DSS Level 1

ANSWER: A

Explanation:

The application might still violate PCI DSS Level 1 even though it is deployed to a Shield Private Space. Heroku Shield Private Spaces provide a platform designed for highly regulated workloads and are part of Heroku's PCI DSS Level 1 Service Provider environment. This gives an application important platform-level controls, including isolated runtime environments, encrypted data services, enhanced logging capabilities, and network controls appropriate for handling cardholder-data workloads.

However, PCI DSS compliance follows a shared-responsibility model. Deploying into a compliant platform does not itself make the client's application, its configuration, business processes, access controls, integrations, or handling of cardholder data compliant. The client remains responsible for implementing and operating the applicable controls in its own scope, such as secure application design, authentication and authorization, secrets management, logging practices, vulnerability management, and limiting storage and exposure of payment data. Compliance must therefore be validated for the complete cardholder-data environment rather than inferred solely from the hosting location. Heroku describes Shield Private Spaces and their compliance capabilities in the Shield Private Spaces documentation; the broader responsibilities for securing an application are covered by Heroku's security documentation.

QUESTION NO: 25

Q. 30 A client has a Heroku Redis instance that can be accessed over the public internet. To meet compliance requirements, the IT Security team requires thatall databases must be isolated from the public internet. Which solution should be suggested by an Architect?

A. Enclose the existing Heroku Redis instance inside a Pnvate Space. Enable Private Space VPN.

B. Use a Heroku Redis instance with a Private Tier plan. Enable Trusted IP ranges on the Redis instance.

C. Use a Heroku Private Space and a Heroku Redis instance with a Private Tier plan. Redeploy the app into the Space

D. Enclose the existing Heroku Redis instance inside a Shield Private Space. Configurethe firewall to allow Redis traffic through

ANSWER: C

Explanation:

Use a Heroku Private Space and a Heroku Redis instance with a Private Tier plan. Redeploy the app into the Space. This design puts both the application and its Redis data store within Heroku's private-network architecture, rather than exposing the database through a public endpoint. Heroku Redis Private Tier is specifically intended for use in Private Spaces and provides a Redis endpoint that is reachable only from applications running in the same Private Space. As a result, application-to-Redis traffic remains on Heroku's private network, satisfying the requirement to isolate the database from the public internet.

Redeploying the application into the Private Space is essential because the app must run in that same space to access the private Redis endpoint. If required, connectivity between the Private Space and the organization's corporate network can be

established through Private Space networking features, such as VPN or peering, without making Redis publicly reachable. This approach combines network isolation with a managed Redis service and is the appropriate Heroku architecture for regulated workloads requiring non-public database access. See Heroku's Heroku Redis documentation and Private Spaces documentation.

QUESTION NO: 26

A client wants to use Heroku Connect to sync data bidirectionally between a Salesforce org and a Heroku Postgres database.

In this scenario, which two are advantages to setting a custom External Identifier field in the Heroku Connect mapping? (Choose two.)

A. External IDs simplify the mapping of polymorphic relationships in the Salesforce org.

B. External IDs improve speed when inserting multiple related objects.

C. External IDs prevent the creation of orphaned duplicates in Heroku Postgres.

D. External IDs are required to enable bidirectional synchronization.

ANSWER: B C

Explanation:

External IDs improve speed when inserting multiple related objects. A custom External Identifier provides a durable business key that Heroku Connect can use while records are being written from Postgres to Salesforce. This is especially useful for related data created in close succession: a child record can reference its parent through the parent's external-ID value rather than requiring the Salesforce-generated record ID to be obtained and propagated first. That reduces coordination and lookup overhead for related write operations.

External IDs prevent the creation of orphaned duplicates in Heroku Postgres. During bidirectional synchronization, the external-ID value gives Heroku Connect a stable way to correlate the local database row with its corresponding Salesforce record. This correlation is valuable when records are created, retried, or subsequently synchronized, because the same logical record can be matched rather than treated as an unrelated new row. A unique, consistently populated external-ID field therefore supports reliable identity resolution across both systems and helps preserve a coherent record relationship as synchronization proceeds.

Heroku Connect mapping configuration and writable-mapping behavior are documented in the Heroku Connect Mappings documentation. Salesforce's use of external IDs for identifying and relating records is covered in the Salesforce Bulk API documentation.

QUESTION NO: 27

Universal Containers has implemented a Knowledge solution for Agents to provide Field Technicians with information necessary to complete assigned work.

Which two capabilities will now be available? (Choose two.)

A. Attach Knowledge Articles to Work Order Line Hems Only.

B. Manage Attached Articles and Search the Knowledge Base

C. include Quick Actions and Global Actions in Attached Articles.

D. Attach Articles to Work Orders and Work Order Line Items.

ANSWER: B D

Explanation:

Manage Attached Articles and Search the Knowledge Base and **Attach Articles to Work Orders and Work Order Line Items.** are available through Salesforce Field Service's integration with Salesforce Knowledge. Knowledge articles can be associated with both work orders and their individual work order line items, allowing personnel to provide guidance at the level that best matches the specific service task. For example, a work order can contain general safety or customer-site instructions, while a line item can have an article describing the repair procedure for a particular asset.

Users working with these records can search Salesforce Knowledge to locate relevant published articles and manage the article relationships from the work order or work order line item. This gives agents an efficient way to curate the instructions that technicians need before or during service execution. Attached articles are then available to technicians in the Field Service mobile experience, helping them access approved troubleshooting steps, procedures, and reference material in the context of their assigned work. Salesforce describes associating knowledge articles with work orders and work order line items in its Field Service Knowledge documentation; the underlying article-management model is covered in Salesforce Knowledge article relationships.

QUESTION NO: 28

A customer's IT department will not allow VPNaccess through their on-premise gateway. Users need to be able to access trusted Heroku applications only when they are in the office. These applications are deployed to a Heroku Private Space.

Which two Private Spaces features should an Architectrecommend to enable on-premise users to gain the access they need? (Choosetwo.)

A. Stable outbound IP addresses

B. VPC peering

C. Internal routing

D. Trusted IP ranges

ANSWER: B D

Explanation:

VPC peering provides private network connectivity between a Heroku Private Space and a customer-managed AWS VPC without creating a VPN connection between those two networks. Where the organization's office network already has connectivity to its AWS environment, such as through existing private network connectivity, peering allows users and systems on that network to reach applications in the Private Space through the private network path. Heroku supports peering connections for Private Spaces, with network ranges configured so that routes do not overlap.

Trusted IP ranges restrict inbound access to applications in a Private Space to requests originating from explicitly approved public IP addresses or CIDR ranges. The organization can add the office's public egress/NAT IP range as a trusted range. Consequently, users receive access while working from the office network, while requests from other untrusted locations are blocked. This combines private connectivity architecture with a clear ingress-access control boundary appropriate for trusted internal applications. See Heroku's Private Space Peering documentation and its Trusted IP Ranges documentation.

QUESTION NO: 29

Which two logs and logging recommendations are prescribed by the Twelve-Factor app methodology? Choose2 answers.

A. Apps should not concernthemselves with routing or storing logs

B. Logs from all processes and backing services should be aggregated together.

C. Logs should be written to a shared file system for scalability.

D. Apps should only keep a limited, rolling buffer of logs available

ANSWER: A B

Explanation:

The Twelve-Factor App methodology treats logs as event streams. An application should not take responsibility for routing, collecting, or persistently storing those streams itself; instead, each process writes its event stream unbuffered to standard output. The execution environment captures that output and can route it to an appropriate destination, such as a log collection system, long-term archive, or real-time analysis service. This separation lets the same application run consistently across local development, staging, and production without embedding environment-specific logging infrastructure.

Aggregating logs from all application processes and backing services is also consistent with the methodology's guidance. A platform or logging service can combine the event streams, giving operators a unified view for querying, monitoring, alerting, and troubleshooting across components. On Heroku, application and platform logs are treated as a unified stream and can be forwarded to external log-management services through log drains. This approach supports scalable, stateless processes while preserving operational visibility. See the Twelve-Factor App: Logs and Heroku Logging documentation.

QUESTION NO: 30

Universal Containers (UC) has a front-end web application and a back-end service application running on Heroku. The applications are running in the Common Runtime. Now, UC wants to prevent any public access to the back-end application. Which two Heroku features should an Architect propose?

A. Private Space VPN Connections

B. Private Spaces DNS Service Discovery

C. Heroku Internal Routing

D. Apache Kafka on Heroku

ANSWER: B C

Explanation:

To remove public access to the back-end application, UC should move the relevant applications into a Heroku Private Space and use Heroku Internal Routing. Internal Routing makes the back-end application reachable only from within its Private Space rather than through the public internet. This is the control that enforces the requirement that the service application not expose a publicly accessible route. The front-end application can then call the service over Heroku's private network.

Private Spaces DNS Service Discovery complements this design by providing internal DNS resolution for applications in the same Private Space. The front end can use the back-end application's internal DNS name to locate and communicate with it using private network connectivity, without depending on a public endpoint. Together, internal-only routing and private DNS discovery provide both the access boundary and the service-to-service addressing needed for this architecture. These capabilities are Private Spaces features, so they are not available while the apps remain solely in the Common Runtime. See Heroku's documentation on Internal Routing and Private Space DNS.

QUESTION NO: 31

A client currently runs a Ruby script in aone-off dyno each time they deploy their Go application to Heroku. The development team wants the script to be executed before the application is deployed because it performs necessary database migrations. Which approach should an Architect recommend?

A. Modify the Go buildpacK to install Ruby, and run the script from the _profile

B. Convert the Ruby release script to Go and execute it on application startup

C. Use both Go and Ruby language buildpacks, and run the Ruby script with release phase

D. Define different process types for the Ruby script and the Go application in the app's Profile.

ANSWER: C

Explanation:

Use both Go and Ruby language buildpacks, and run the Ruby script with release phase is the appropriate approach because the application requires dependencies from two language ecosystems: Go for the application itself and Ruby for the migration script. Heroku supports multiple buildpacks, allowing each buildpack to detect and prepare the relevant part of the application during the build. The Ruby buildpack makes the Ruby runtime and dependencies available, while the Go buildpack builds the Go application.

A `release` process in the app's `Procfile` is specifically intended for one-time tasks that must run during a release, such as database migrations. Heroku runs the release command in a one-off dyno after the slug is built but before the release is made available to web and worker dynos. If the migration task fails, the release does not proceed, helping ensure that the newly deployed application is not started against an incompatible database schema. This keeps deployment orchestration within Heroku's release lifecycle rather than relying on a separately triggered manual dyno.

See Heroku's documentation for using multiple buildpacks and the release phase.

QUESTION NO: 32

A client wants to perform complex processing on large data sets in Salesforce. The data is confidential, but it does not have advanced compliance requirements.

What Heroku Enterprise features should an Architect recommend in this scenario?

A. Shield Private Spaces and Heroku External Objects

B. Private Spaces and Heroku Connect

C. Private Spaces and Heroku External Objects

D. Shield Private Spaces and Heroku Shield Connect

ANSWER: B

Explanation:

Private Spaces and Heroku Connect is the appropriate recommendation because it combines a dedicated Heroku runtime environment with managed, bidirectional data synchronization between Salesforce and a Heroku Postgres database. Heroku Connect is designed for applications that need to work with Salesforce data in a relational database, where the application can use SQL, background workers, and Heroku's compute resources to process large volumes of data without forcing all processing through Salesforce's multitenant transaction limits. It can synchronize selected Salesforce objects to Postgres and optionally write supported changes back to Salesforce, allowing the architecture to keep Salesforce as a system of record while performing computationally intensive work externally.

Private Spaces provide an isolated network environment for the application and can support private connectivity patterns appropriate for confidential business data. Since the scenario explicitly says that advanced compliance requirements are not needed, the standard Private Spaces offering supplies the relevant isolation and enterprise networking foundation without requiring Shield-specific capabilities. The solution therefore aligns data-processing needs, confidentiality, and enterprise deployment controls while avoiding unnecessary compliance-focused features. See the Heroku Connect documentation and Heroku Private Spaces documentation.

QUESTION NO: 33

Which three overview cards does the Field Service mobile app provide as context to Technicians on upcoming Service Appointments? (Choose three.)

A. Asset History

B. Contact

C. Address

D. Site Details

E. Product Catalog

ANSWER: B C D

Explanation:

The Field Service mobile app presents **Contact**, **Address**, and **Site Details** overview cards to give technicians essential context before and while completing a service appointment. The Contact card surfaces the relevant customer contact information, enabling the technician to identify and communicate with the person associated with the work. The Address card provides the service location information needed for travel and arrival at the appointment. The Site Details card gives technicians useful information about the service site, helping them understand the location-specific context before work begins.

These cards are intended to make critical appointment information immediately accessible in the mobile workflow, reducing the need for technicians to navigate through related records when preparing for an upcoming job. Administrators can configure the Field Service mobile experience to ensure that the information technicians need is available in the appropriate mobile context. See Salesforce's Customize the Field Service Mobile App Trailhead module and the Field Service Mobile App documentation for mobile-app configuration and usage details.

QUESTION NO: 34

Universal Containers provides installation, repair, and consulting services When Technicians complete the work, they need to providedifferent reports for the installation, repair, and consulting services.

Which two configurations should a Consultant recommend to meet this requirement? (Choose two)

A. Work Types

B. Assets

C. Service Report Templates

D. Product Templates

ANSWER: A C

Explanation:

Work Types and **Service Report Templates** provide the needed configuration. A work type defines a standardized category of field service work, such as installation, repair, or consulting. It can supply default information and processes for work orders and service appointments created for that type of job. Creating separate work types therefore lets Universal Containers consistently distinguish the work technicians perform and apply the appropriate service-process defaults.

Service report templates define the document technicians generate after completing work. Each template can include different sections, fields, questions, branding, and related-record information, allowing the organization to produce an installation report with installation-specific details, a repair report with diagnostic and resolution information, and a consulting report focused on advisory work. Field Service supports associating the appropriate report-template experience with the work being performed, so technicians can generate a report suited to the service delivered rather than using one generic document for every visit.

Together, these configurations standardize the service categories and ensure completed work produces the appropriate customer-facing documentation. See Salesforce's Work Types documentation and Service Report Templates documentation.

QUESTION NO: 35

A client is moving a customer-facing application to Heroku. The application must perform a database backup before a planned, high-risk data migration, and the team must be able to restore the data if the migration fails.

Which approach should an Architect recommend?

A. Create a Heroku Postgres logical backup before the migration and validate the restoration procedure in a nonproduction environment.

B. Create a new Dataclip that selects all rows from every production table.

C. Scale the web dynos to zero after the migration has completed successfully.

D. Fork the database after making the production schema changes.

ANSWER: A

Explanation:

Create a Heroku Postgres logical backup before the migration and validate the restoration procedure in a nonproduction environment. A backup provides a recoverable copy of the database that can be restored if the migration has an unexpected result. Testing restoration before the production change confirms that the team understands the recovery process and the expected recovery time.

A follower database is intended for replication and disaster-recovery scenarios, but it is not a replacement for a deliberate pre-migration backup. The team should also plan the migration window and communicate any expected application impact. See Heroku's Heroku Postgres Backups documentation.

QUESTION NO: 36

Heroku add-ons are cool because they:

A. Give you the ability to add complex functionality to your application without having to manage the underlying software or infrastructure.

B. Accessorize your dynos with over-clocked CPU power and faster I/O speeds

C. Can be provisioned easily from the Heroku Enterprise Flea Market system.

D. Cost nothing — billing is easy because they're always provided at no additional cost with every Heroku Enterprise account.

ANSWER: A

Explanation:

Heroku add-ons provide managed, integrated services that applications can use without the development team having to install, operate, patch, scale, or monitor the underlying service infrastructure themselves. An add-on can supply capabilities such as data storage, logging, monitoring, caching, messaging, email delivery, or other operational features. It is provisioned for a particular Heroku app, and Heroku configures the credentials or connection information as application configuration variables, allowing the app to consume the service through its normal runtime configuration. This model lets teams add production-ready capabilities quickly while keeping their focus on application code and delivery rather than service administration. Add-ons are offered through the Heroku Elements marketplace and can have free, paid, or usage-based plans depending on the provider and plan selected. Heroku’s documentation describes add-ons as cloud services that provide extended functionality for an application and explains that provisioning makes the relevant configuration available to the app. See the Heroku Add-ons documentation and the guide to provisioning Heroku add-ons.