Oracle.1z0-1084-24.v2025-02-04.q34

試験コード:	1z0-1084-24
試験名称:	Oracle Cloud Infrastructure 2024 Developer Professional
認定資格:	Oracle
無料問題数:	34
バージョン:	v2025-02-04
アクセス数:	448
ページビュー数:	340
https://www.jpnpdf.com/Oracle.1z0-1084-24.v2025-02-04.q34-mondaishu.html	

最新問題: **1**
マイクロサービスを最もよく定義するオプションはどれですか?
A. 静的に型付けされコンパイルされる言語。
B. コンテナ化されたアプリケーションの展開、スケーリング、管理を自動化するオープン ソース システム。
C. 単一または少数のタスクを実行する、細かく調整されたソフトウェア。
D. 構造化された情報またはデータの整理されたコレクション。通常はコンピュータ システムに電子的に保存されます。
Answer: C (メーッセージを残す)
正解は、 単一または少数のタスクを実行する、細かく調整されたソフトウェア」です。マイクロサービスは、システムを特定の一連のタスクを実行する小さな独立したサービスに分解するソフトウェア アーキテクチャ アプローチです。各マイクロサービスは、単一または少数の関連タスクを処理するように、集中的かつ細かく調整され、非常にまとまりのあるように設計されています。この粒度により、複雑なアプリケーションを構築する際のスケーラビリティ、保守性、柔軟性が向上します。提供されているその他のオプションは、マイクロサービスを正確に定義していません。コンテナ化されたアプリケーションの展開、スケーリング、管理を自動化するオープン ソース システムは、Kubernetes などのコンテナ オーケストレーション ツールを指します。これは、マイクロサービスの管理に使用できますが、マイクロサービス自体の定義ではありません。静的に型付けおよびコンパイルされた言語は、プログラミング言語の特性のタイプを表すものであり、マイクロサービスの概念に固有のものではありません。構造化された情報またはデータの整理されたコレクション (通常はコンピューター システムに電子的に保存されます) は、データベースまたはデータ ストレージ システムの定義であり、マイクロサービスとは無関係です。

最新問題: **2**

OKE クラスターを、攻撃対象領域が最も少ないネットワーク構成に更新し、同時にデプロイされたアプリケーションにインターネットから直接アクセスできるようにするように求められています。この要件を満たす有効な OKE クラスター ネットワーク構成はどれですか? (最適な回答を選択してください。)
A. ノード、Kubemetes API エンドポイント、ロードバランサーのプライベートサブネット
B. ノード用のプライベートサブネット、Kubemetes APIエンドポイントとロードバランサー用のパブリックサブネット
C. ノードと Kubemetes API エンドポイント用のプライベートサブネット、ロードバランサー用のパブリックサブネット
D. Kubemetes API エンドポイントのプライベートサブネット、ノードとロードバランサーのパブリックサブネット
Answer: C (メーッセージを残す**)**
攻撃対象領域を最小限に抑えながら、インターネットからデプロイされたアプリケーションへの直接アクセスを可能にするという要件を満たす有効な OKE クラスター ネットワーク構成は、ノードと Kubernetes API エンドポイント用のプライベート サブネット、ロードバランサー用のパブリック サブネットです。ノードと Kubernetes API エンドポイントをプライベート サブネットに配置すると、インターネットから直接アクセスできなくなり、攻撃対象領域が縮小されます。
一方、ロード バランサはパブリック サブネットに配置され、インターネットからアクセスでき、デプロイされたアプリケーションにアクセスするためのエントリ ポイントとして機能します。この構成により、ノードや API エンドポイントなどのクラスターの重要なコンポーネントがプライベート ネットワーク内で保護され、ロード バランサを介してアプリケーションへのアクセスが提供されます。これにより、デプロイされたアプリケーションの可用性を維持しながら、クラスターの内部コンポーネントへの直接アクセスを制限することで、セキュリティが強化されます。

最新問題**: 3**
Docker イメージは 1 つ以上のレイヤーで構成され、各レイヤーは Dockerfile 命令を表します。レイヤーは積み重ねられ、各レイヤーは前のレイヤーからの変更のデルタです。これらのレイヤーにはどのような権限が関連付けられていますか?
A. 主に読む
B. 書き込みのみ
C. 移動可能
D. 読み取り専用
E. 一度だけ書き込む
Answer: (解答を表示する**)**
正解は 「読み取り専用」です。Docker イメージのレイヤーは読み取り専用です。レイヤーが作成されると、変更することはできません。各レイヤーは Dockerfile 命令を表し、前のレイヤーの上に積み重ねられ、不変のレイヤーのスタックを形成します。これらのレイヤーは、イメージの一貫性と整合性を確保するために読み取り専用になるように設計されていま

す。Docker イメージがビルドされると、Dockerfile 内の各命令によって新しいレイヤーが作成されます。各レイヤーは、前のレイヤーに対するその命令による変更を表します。レイヤーは互いに積み重ねられ、完全なイメージを形成します。このレイヤーベースのアプローチにより、Docker イメージを効率的に保存および配布できます。レイヤーは読み取り専用であるため、イメージに変更または修正を加えると、既存のレイヤーが変更されるのではなく、新しいレイヤーが作成されます。この不変性により、各レイヤーがそのまま残り、イメージの整合性が維持されます。また、Docker のキャッシュ メカニズムが有効になり、対応する命令が変更されていない場合は、以前にビルドされたレイヤーを再利用できるため、イメージ ビルド プロセスが高速化されます。 書き込みのみ」、⇈ 回だけ書き込み」、 移動可能」、 ほとんど読み取り」などの他のオプションは、Docker イメージ レイヤーに関連付けられた権限を正確に表すものではありません。Docker イメージ レイヤーは、読み取り専用として特別に設計されています。

最新問題: **4**
Docker イメージとコンテナに関して正しい記述はどれですか? (2 つ選択してください。)
A. 特定のイメージから一度に生成できるコンテナーは 1 つだけです。
B. イメージは不変のレイヤーの集合ですが、コンテナはイメージの実行中のインスタンスです。
C. 同じイメージから複数のコンテナが生成された場合、それらはすべてメモリ内でそのイメージの同じコピーを使用します。
D. 新しい Dockerfile を記述してビルドすることが、新しい Docker イメージを作成できる唯一の方法です。
E. コンテナはイメージがなくても存在できますが、イメージはコンテナなしでは存在できません。
Answer: B,C (メーッセージを残す**)**
Docker イメージとコンテナに関する正しい記述は次のうちどれですか。(2 つ選択してください。) 新しい Dockerfile を記述して構築することが、新しい Docker イメージを作成する唯一の方法です。コンテナはイメージがなくても存在できますが、イメージはコンテナなしでは存在できません。特定のイメージから一度に生成できるコンテナは 1 つだけです。同じイメージから複数のコンテナが生成された場合、それらはすべてメモリ内でそのイメージの同じコピーを使用します。(正解) イメージは不変のレイヤーのコレクションですが、コンテナはイメージの実行中のインスタンスです。(正解) ChatGPT Docker イメージとコンテナに関する正しい記述は次のとおりです。同じイメージから複数のコンテナが生成された場合、それらはすべてメモリ内でそのイメージの同じコピーを使用します。この記述は正解です。同じ Docker イメージから複数のコンテナが作成されると、それらはメモリ内でイメージの基になるレイヤーを共有します。各コンテナには独自の分離されたファイルシステムとランタイム環境がありますが、それらはすべて同じベースイメージを使用するため、リソースを効率的に使用できます。イメージは不変のレイヤーのコレクションですが、コンテナはイメージの実行中のインスタンスです。この記述も正しいです。Docker

イメージは複数のレイヤーで構成され、各レイヤーは前のレイヤーに対する特定の変更または追加を表します。これらのレイヤーは読み取り専用で、複数のコンテナー間で共有できます。一方、コンテナーは特定のイメージから作成された軽量で分離されたランタイム インスタンスです。これは、独自のファイル システム、ネットワーク、およびランタイム構成を持つ実行中のプロセスです。 新しい Dockerfile を記述してビルドすることが、新しい Docker イメージを作成する唯一の方法です」および 「コンテナーはイメージがなくても存在できますが、イメージはコンテナーなしでは存在できません」という記述は正しくありません。Docker イメージは、Dockerfile の使用、既存のイメージからのインポート、レジストリからのプルなど、さまざまな方法で作成できます。さらに、コンテナーを実行するにはイメージが必要なので、イメージが存在しなければコンテナーを作成することはできません。

最新問題: **5**
kubectl CLI から Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE) クラスタにアクセスするために必要なものはどれですか? (2 つ選択してください。)
A. OKE クラスターで Tiller が有効になっています。
B. クラスター ワーカー ノードに追加された公開キーを含む SSH キー ペア。
C. OCI CLI をインストールして構成します。
D. 構成された OCI API 署名キー ペア。
E. OCI アイデンティティおよびアクセス管理 (IAM) 認証トークン。
Answer: C,D (メーッセージを残す**)**
正しいオプションは次のとおりです: 構成された OCI API 署名キー ペア: API 署名キー ペアは、OKE クラスターを含む OCI リソースにアクセスするための認証と承認に使用されます。API リクエストを認証するには、ローカル マシンで秘密キーを構成する必要があります。クラスター ワーカー ノードに追加された公開キーを含む SSH キー ペア: これは、OKE クラスター内のワーカー ノードへの安全な SSH アクセスに必要です。
クラスターの作成または更新中に、SSH キー ペアを生成し、公開キーをクラスターのワーカー ノード プールに追加する必要があります。したがって、正しいオプションは、構成された OCI API 署名キー ペアと、公開キーがクラスター ワーカー ノードに追加された SSH キー ペアを持つことです。

最新問題: **6**
myapp アプリケーションの myfunction という名前の Cloud Shell セッションから Oracle Functions を呼び出すときに、予期しないエラーが発生しました。どのオプションを選択すると、エラーに関する詳細情報を取得できますか。
A. fn --verbose 呼び出し myapp myfunction
B. fn --debug 呼び出し myapp myfunction
C. エラーメッセージをOracleサポートに連絡してください
D. DEBUG=1 fn 呼び出し myapp myfunction

Answer: D (メーッセージを残す)
Cloud Shell セッションから Oracle Functions を呼び出すときにエラーに関する詳細情報を取得するオプションは、「DEBUG=1 fn invoke myapp myfunction」です。fn コマンドを使用して関数を呼び出す前に環境変数 DEBUG=1 を設定すると、デバッグ モードを有効にして、関数の実行に関する詳細情報を取得できます。これは、トラブルシューティングやエラーの根本原因の把握に役立ちます。コマンド「DEBUG=1 fn invoke myapp myfunction」を使用すると、デバッグ モードを有効にして関数呼び出しが実行され、追加のデバッグ情報がコンソール出力に表示されます。この情報には、スタック トレース、詳細なエラー メッセージ、および問題の特定と解決に役立つその他の関連情報が含まれます。fn コマンドで詳細オプション (--verbose) またはデバッグ オプション (--debug) を使用すると、追加情報が提供される場合もありますが、具体的な動作は fn CLI ツールのバージョンと構成によって異なる場合があります。エラー メッセージとともに Oracle サポートに連絡することも常に可能ですが、DEBUG=1 環境変数を使用してデバッグ モードを有効にすると、より詳細な情報にすぐにアクセスできるようになり、エラーをより効率的に診断および解決できるようになります。

最新問題**: 7**
Oracle Cloud Infrastructure (OCI) API Gateway デプロイメントを構成するときに使用できる有効なバックエンド タイプ オプションではないのはどれですか。
A. HTTP_BACKEND
B. ORACLE STREAMS_BACKEND
C. ORACLE_FUNCTIONS_BACKEND
Answer: B (メーッセージを残す)
OCI API Gateway デプロイメントを構成するときは、API デプロイメント仕様 3 で各ルートのバックエンド タイプを指定する必要があります。バックエンド タイプによって、API ゲートウェイがそのルートへのリクエストを処理し、適切なバックエンド サービス 3 に転送する方法が決まります。OCI API Gateway デプロイメント 3 で有効なオプションは、次のバックエンド タイプです。
* HTTP_BACKEND: API ゲートウェイは、バックエンド サービスとして HTTP または HTTPS URL にリクエストを転送します。
* ORACLE_FUNCTIONS_BACKEND: API ゲートウェイは、バックエンド サービスとして Oracle Functions 関数を呼び出します。
* STOCK_RESPONSE_BACKEND: API ゲートウェイは、バックエンド サービスを呼び出さずに標準レスポンスを返します。ORACLE STREAMS_BACKEND は、OCI API Gateway デプロイメントの有効なバックエンド タイプではありません。Oracle Streams は、大量のデータをリアルタイムで取り込み、消費するために使用できる、完全に管理されたスケーラブルで耐久性のあるメッセージング サービスです4。ただし、Oracle Streams は、OCI API Gateway デプロイメントのバックエンド サービスとしてはサポートされていません。

最新問題: **8**
Oracle Cloud Infrastructure (OCI) にデプロイされる車両群のリアルタイム監視アプリケーションを開発しています。車両からのリアルタイム データ フィードを処理するには、OCI キューと OCI ストリーミングのどちらを使用するかを選択する必要があります。説明されているシナリオに基づくと、リアルタイム データ フィードを処理するための最も適切な選択肢はどれですか。
A. OCIストリーミングは、大量の継続的なデータの取り込みと処理向けに設計されているため、車両群に最適な選択肢となります。
B. OCIストリーミング。リアルタイムアプリケーションに必要な、1回限りのメッセージ配信を提供するため
C. OCI キュー。低遅延メッセージングに最適化されており、リアルタイムアプリケーションに最適です。
D. OCI キュー。リアルタイム監視アプリケーションにとって重要な、少なくとも 1 回のメッセージ配信を提供するため
Answer: A (メーッセージを残す**)**
OCI ストリーミングは、リアルタイムで消費および処理できる、継続的で大量のデータ ストリームを取り込むための、フル マネージドでスケーラブルかつ耐久性に優れたメッセージング ソリューションです1。ストリーミングは、パブリッシュ/サブスクライブ メッセージング モデル1 でデータが継続的かつ順次生成および処理されるあらゆるユース ケースに適しています。ストリーミングは、低レイテンシで 1 秒あたり数百万のメッセージを処理できます2。したがって、OCI ストリーミングは、車両群からのリアルタイム データ フィードを処理するのに最適な選択肢です。検証済みの参照: ストリーミングの概要、Container Engine for Kubernetes

最新問題: **9**
Oracle Cloud Infrastructure Registry (OCIR) からクライアント マシンに Docker イメージを取得するには、どのコマンドを使用しますか?
A. docker pull <リージョンキー>.ocir.io/<テナンシー名前空間>/<リポジトリ名>: <タグ>
B. docker pull <テナンシー名前空間>/<リージョンキー>.ocir.io/<リポジトリ名>: <タグ>
C. docker fetch <リージョンキー>.ocir.io/<テナンシー名前空間>/<リポジトリ名>:<タグ>
D. docker fetch <テナンシー名前空間>/<リージョンキー>.ocir.io/<リポジトリ名>:<タグ>
Answer: A (メーッセージを残す**)**
OCI レジストリからクライアント マシンに Docker イメージをプルするには、次の構文1 で docker pull コマンドを使用する必要があります: docker pull <region-key>.ocir.io/<tenancy-namespace>/<repo-name>:<tag> ここで、
* <region-key> は、使用している OCI レジストリ リージョンのキーです。たとえば、iad です。リージョン 1 の可用性を参照してください。
* ocir.io は OCI レジストリ名です。

* <tenancy-namespace> は、イメージをプルするリポジトリを所有するテナンシーの自動生成されたオブジェクト ストレージ名前空間文字列です (テナンシー情報ページに表示されます)1。
* <repo-name> は、プルするイメージが含まれているリポジトリの名前です。
* <tag> は、取得するイメージのタグです。

最新問題: **10**
(CHK_4>2) Oracle Cloud Infrastructure (OCI) ストリーミング サービスに関して有効でない記述はどれですか (2 つ選択してください)。
A. OCI ストリーミングは、デフォルトですべてのデータを 24 時間保存しますが、最大 7 日間まで延長できます。B
B. OCI ストリーミングは転送中のすべてのデータを自動的に暗号化しますが、必要に応じて保存中のデータを暗号化するのは開発者の責任です。
C. ストリームのスループットはパーティションによって定義されます。パーティションは1MB/秒のデータ入力と2MBの
/秒のデータ出力。
D. ストリームは、顧客管理の暗号化キーをサポートするパブリック エンドポイントまたはプライベート エンドポイントのいずれかで構成できます。
E. OCI ストリーミングは、各パーティションに対して 1 秒あたり最大 2,000 件のリクエストをサポートできます。
Answer: D,E (メーッセージを残す**)**
Oracle Cloud Infrastructure (OCI) ストリーミング サービスに関して、有効でない 2 つの記述は次のとおりです。ストリームは、顧客管理の暗号化キーをサポートするパブリック エンドポイントまたはプライベート エンドポイントのいずれかで構成できます。OCI ストリーミング サービスは現在プライベート エンドポイントのみをサポートしているため、この記述は無効です。顧客管理の暗号化キーは現在、OCI ストリーミングではサポートされていません。OCI ストリーミングは、各パーティションに対して 1 秒あたり最大 2,000 件のリクエストをサポートできます。ストリームのスループットは、1 秒あたりのリクエスト数という観点からパーティションによって定義されないため、この記述は無効です。ストリームのスループットは、データの入出力速度という観点から定義されます。各パーティションは 1 MB/秒のデータ入力と 2 MB/秒のデータ出力を提供しますが、1 秒あたりのリクエスト数という特定の数には対応していません。その他の記述は有効です。OCI ストリーミングは、デフォルトですべてのデータを 24 時間保存しますが、最大 7 日間まで延長できます。OCI ストリーミングは転送中にすべてのデータを自動的に暗号化しますが、必要に応じて保存データを暗号化するのは開発者の責任です。

最新問題: **11**
Python アプリケーションを Oracle Container Engine for Kubernetes (OKE) クラスタにデプロイしました。ただし、テスト中にバグが見つかったため、修正して新しい Docker イ

メージを作成しました。次に、この新しいイメージをデプロイした後に動作しない場合は、以前のバージョンにロールバックできるようにする必要があります。kubect1 を使用する場合、どの戦略を使用すればよいでしょうか。
A. ブルー/グリーンデプロイメント
B. カナリアデプロイメント
C. ローリングアップデート
D. A/B テスト
Answer: (解答を表示する**)**
ローリング アップデートは、ダウンタイムなしでアプリケーションの古いバージョンを新しいバージョンに徐々に置き換えるデプロイメント戦略です4。OKE は、kubectl rollout コマンド4 を使用してローリング アップデートをサポートします。ローリング アップデートを使用すると、新しいバージョンで問題が発生した場合に、以前のバージョンにロールバックできます4。
したがって、kubectl でローリング アップデート戦略を使用すると、新しいイメージがデプロイ後に機能しない場合でも、Python アプリケーションの以前のバージョンにロールバックできるようになります。検証済みの参照: Oracle Container Engine for Kubernetes のデプロイ

最新問題**: 12**
あなたのチームは、OCI Container Engine for Kubernetes (OKE) クラスタ内のマイクロサービス デプロイメントに関連付けられた Kubernetes シークレットを暗号化するために、Oracle Cloud Infrastructure (OCI) Vault 内でマスター暗号化キー (MEK) を使用することを選択しました。これにより、キーのローテーションを簡単に管理できます。OCI Vault サービスでのキーのローテーションに関して、次のうち正しくないものはどれですか。
A. ローテーション後は、古いキー バージョンは削除されるまで暗号化に使用できます。
B. ソフトウェアと HSM で保護された MEKS の両方をローテーションできます。
C. MEK をローテーションすると、新しいキー バージョンが自動的に生成されます。
D. 各キー バージョンは、個別の一意の OCIDS を使用して内部的に追跡されます。
Answer: A (メーッセージを残す**)**
正解は、 ローテーション後は、古いキー バージョンは削除されるまで暗号化に使用できます」です。OCI Vault サービスでのキーのローテーションに関して正しくない記述は、 ローテーション後は、古いキー バージョンは削除されるまで暗号化に使用できます」です。OCI Vault サービスでは、マスター暗号化キー (MEK) をローテーションすると、新しいキー バージョンが自動的に生成されます。ただし、キーがローテーションされて新しいバージョンが作成されると、古いキー バージョンは暗号化に使用できなくなります。キー ローテーションの目的は、暗号化キーが定期的に更新され、古いキーが機密データの保護に使用されないようにすることです。これにより、潜在的なキー侵害の影響を最小限に抑えてセキュリティが強化されます。その他の記述は有効です。ソフトウェア セキュリティ モジュール (HSM) で保護された MEK とハードウェア セキュリティ モジュール (HSM) で保

護された MEK の両方をローテーションできます。これにより、MEK のタイプを柔軟に選択でき、使用する暗号化方法に関係なくキー ローテーションを実行できます。各キー バージョンは、個別の一意の OCID (Oracle Cloud 識別子) を使用して内部的に追跡されます。これにより、OCI Vault サービス内のさまざまなキー バージョンを簡単に管理および追跡できます。要約すると、有効ではないステートメントは、古いキー バージョンが削除されるまで暗号化に引き続き使用できることを示唆するステートメントです。キー ローテーションは、最新のキー バージョンの使用を保証し、古いキー バージョンを廃止してセキュリティを強化するように設計されています。

最新問題**: 13**
組織では、Oracle Container Engine for Kubernetes (OKE) に電子商取引アプリケーションをデプロイし、Docker イメージ リポジトリとして Oracle Cloud Infrastructure Registry (OCIR) サービスを使用しています。OKE クラスタは ゐスタム作成」オプションを使用してデプロイされており、Virtual Cloud Network (VCN) には、ルート テーブル、セキュリティ リスト、インターネット ゲートウェイが関連付けられた 3 つのパブリック サブネットがあります。
しかし、アプリケーション コンテナのデプロイに失敗しています。調査してみると、YAML 構成にイメージへの正しいパスがあるにもかかわらず、指定された OCIR リポジトリからイメージがプルされていないことがわかりました。さらに調査する必要がある有効な懸念事項は何でしょうか。
A. OCIR サービスに接続するには、TCP ポート 22 のセキュリティ リスト ルールを追加する必要があります。
B. OKE クラスタ ワーカー ノードをホストする VCN には、OCIR リポジトリにアクセスするための NAT ゲートウェイが必要です。
C. OKE クラスターにアプリケーションをデプロイするユーザーごとに、Identity and Access Management (IAM) 認証情報を追加する必要があります。
D. OKE クラスターには、OCIR リポジトリの資格情報を含むシークレットが必要であり、そのシークレットを Kubernetes デプロイメント マニフェストで使用する必要があります。
Answer: (解答を表示する**)**
このシナリオでさらに調査する必要がある正当な懸念事項は、OKE クラスターに Oracle Cloud Infrastructure Registry (OCIR) リポジトリの資格情報を含むシークレットがあるかどうか、およびそのシークレットが Kubernetes デプロイメント マニフェストで使用されているかどうかです。この懸念事項が関連する理由は次のとおりです。OCIR リポジトリへのアクセス: OKE クラスターが OCIR リポジトリからイメージをプルするには、適切な認証資格情報が必要です。これらの資格情報は通常、レジストリでの認証に必要な情報を含むシークレットの形式で提供されます。デプロイメント マニフェスト内のシークレット: Kubernetes デプロイメント マニフェストは、アプリケーション コンテナのデプロイ方法を定義します。これには、コンテナ イメージ、リソース要件、環境変数などの仕様が含ま

れます。OCIR などのプライベート リポジトリからイメージをプルするには、デプロイメント マニフェストで、レジストリ資格情報を含む適切なシークレットを参照する必要があります。指定された OCIR リポジトリからイメージがプルされていない場合は、OCIR 資格情報を含むシークレットが欠落しているか、デプロイメント マニフェストで適切に参照されていないことが示唆されます。さらなる調査では、シークレットの存在と正確さを検証すること、およびアプリケーション コンテナのデプロイメント マニフェストでシークレットが正しく参照されていることを確認することに重点を置く必要があります。デプロイメント マニフェストにシークレットが存在し、適切な構成であることを確認することで、OKE クラスターは OCIR リポジトリにアクセスし、アプリケーション コンテナを正常にデプロイするために必要な資格情報を持つことになります。

最新問題: **14**
Kubernetes には、コンピューティング、ネットワーク、ストレージなどのさまざまな要素が含まれます。コンピューティングは基本的に CPU (ユニット) とメモリ (バイト) です。OKE クラスター内では、コンピューティングに関するデプロイメントの最小単位とは何ですか?
A. コンテナ
B. サービス
C. ポッド
D. 名前空間
E. デプロイメントリソース
Answer: (解答を表示する**)**
ポッドは、Kubernetes オブジェクト モデルで作成またはデプロイする最小かつ最も単純な単位です2。ポッドは、クラスターで実行中のプロセスの 1 つのインスタンスを表します。ポッドには、Docker コンテナーなどの 1 つ以上のコンテナーが含まれます。ポッドを作成するときに、各コンテナーに必要な CPU とメモリ (RAM) の量を定義します。ポッドには、ストレージ ボリューム、IP アドレス、コンテナーの実行方法を制御するオプションなどを含めることもできます2。ポッドは、デプロイメント、レプリカ セット、サービスなどの大規模な Kubernetes 構成要素の基本的な構成要素です2。

最新問題: **15**
マイクロサービスを開発する場合、それぞれを任意の言語で開発できます。このタイプの開発を説明する用語は何ですか? (最適な回答を選択してください。)
A. アジャイル
B. DevOps
C. 分散型
D. ポリグロット
Answer: C (メーッセージを残す**)**
さまざまな言語を選択してマイクロサービスを開発することを ポリグロット」と呼びます。ポリグロット アーキテクチャでは、各マイクロサービスは、特定の要件に最も適したプ

ログラミング言語またはテクノロジ スタックを使用して開発されます。このアプローチにより、開発者はさまざまな言語とフレームワークの長所を活用でき、サービス間の相互運用性を維持しながら、各マイクロサービスに最も適したツールを使用できます。

最新問題: **16**
Container Engine for Kubernetes (OKE) クラスタを作成したら、Oracle Cloud Infrastructure (OCI) Logging を使用して、クラスタ内のワーカー ノード コンピュート インスタンスで実行されているアプリケーションのログを表示および検索できます。アプリケーション ログを収集して解析するために必要のないタスクはどれですか? (最適な回答を選択してください。)
A. クラスター内のすべてのワーカー ノードを含むルールを使用して動的グループを作成します。
B. クラスターの OCI ロギング オプションを [有効] に設定します。
C. クラスター内のすべてのワーカーノードの監視を有効にします。
D. 適切なエージェント構成を使用して、OCI Logging でカスタム ログを構成します。
Answer: C (メーッセージを残す**)**
正解は、クラスタ内のすべてのワーカー ・ノードの監視を有効にします。Oracle Cloud Infrastructure (OCI) Loggingを使用してアプリケーション ・ログを収集および解析するために、クラスタ内のすべてのワーカー ・ノードの監視を有効にする必要はありません。監視は、メトリックを収集し、ワーカー ・ノードのヘルスとパフォーマンスを監視できる別の機能です。アプリケーション ・ログを収集および解析するには、次のタスクを実行する必要があります。クラスタのOCI Loggingオプションを 「有効」に設定する: これにより、クラスタのOCI Loggingサービスが有効になります。クラスタ内のすべてのワーカー ・ノードを含むルールを持つ動的グループを作成する: これにより、ワーカー ・ノードによって生成されたログをターゲットにすることができます。適切なエージェント構成を使用してOCI Loggingでカスタム ・ログを構成する: これには、アプリケーション ・ログを解析および収集するためのログ ・ソース、ログ ・パス、およびログ形式を指定することが含まれます。これらのタスクを完了することで、OKEクラスタ内のワーカー ・ノード ・コンピュート ・インスタンスで実行されているアプリケーションによって生成されたアプリケーション ・ログを収集および解析できます。

最新問題: **17**
Java、Python、Node.js で複数の Oracle Functions を使用して、クラウド ネイティブのサーバーレス旅行アプリケーションを構築しています。これらの関数を構築して、travel-app という単一のアプリケーションにデプロイする必要があります。このタスクを正常に完了するには、どのコマンドが役立ちますか。
A. fn 関数 deploy app travel-app--all
B. fn app deploy --app travel-app --all
C. fn app --app travel-app deploy --ext java pyljs
D. fn deploy--app travel-app --all
Answer: D (メーッセージを残す**)**
正解は、fn deploy --app travel-app --all です。説明: 複数の Oracle Functions を 「travel-app」という単一のアプリケーションの一部としてビルドしてデプロイするには、適切なオプションを指定した fn deploy コマンドを使用します。コマンド fn deploy --app travel-app --all が正しい構文です。コマンドの各部分の動作は次のとおりです。fn deploy: このコマンドは、Oracle Functions で関数とアプリケーションをデプロイするために使用されます。--app travel-app: このオプションは、アプリケーション名を 「travel-app」として指定し、このアプリケーションに関数をデプロイすることを示します。--all: このオプションは、アプリケーション内のすべての関数をデプロイすることを示します。fn deploy --app travel-app --all を使用すると、さまざまなプログラミング言語 (Java、Python、Node.js) にわたる旅行アプリケーション内のすべての関数を Oracle Functions の 「travel-app」アプリケーションにビルドしてデプロイできます。

最新問題: **18**
Oracle Cloud Infrastructure (OCI) API Gateway で API デプロイメントを作成し、アクセスを制御するためのリクエスト ポリシーを構成する必要があります。OCI API Gateway で使用できないのはどれですか。
A. バックエンド OCI リソースへのアクセスを制御します。
B. バックエンド サービスに送信されるリクエストの数を制限します。
C. クロスオリジンリソース共有 (CORS) サポートを有効にします。
D. 認証と承認を提供します。
Answer: (解答を表示する**)**
正解は、バックエンド OCI リソースへのアクセスの制御です。OCI API Gateway は、バックエンド OCI リソースへのアクセスを直接制御しません。主に、ゲートウェイを通じて公開される API へのアクセスの管理とセキュリティ保護に重点を置いています。ゲートウェイは API のフロントエンドとして機能し、認証、承認、レート制限、CORS サポートなどの機能を提供します。OCI API Gateway では、認証および承認ポリシーを構成し、リクエスト

数を制限し、CORS サポートを有効にすることができますが、バックエンド OCI リソースへのアクセスを直接制御することはできません。バックエンド リソースへのアクセスは通常、IAM ポリシー、ネットワーク セキュリティ ルール、リソース固有のアクセス制御などの他の手段によって管理されます。

最新問題: **19**
次のどれが、マイクロサービスによって通常満たされる基準ではありませんか?
A. ビジネス機能を中心に構成されています。
B. 密結合
C. メンテナンス性が高い
D. 独立して展開可能
Answer: B (メーッセージを残す**)**
正解は、「密結合」です。密結合は、マイクロサービスが通常満たす基準ではありません。実際、マイクロサービスは疎結合になるように設計されています。疎結合とは、依存関係を減らし、異なるコンポーネントまたはサービス間の直接的なやり取りを最小限に抑えることを指します。マイクロサービスは独立性と自律性を促進し、各サービスが他のサービスに密結合されることなく独立して動作できるようにします。リストされているその他のオプションは、マイクロサービスが通常満たす基準です。
ビジネス機能を中心に構成: マイクロサービス アーキテクチャでは、特定のビジネス機能または機能を中心にサービスを設計することを提案しています。これにより、組織のビジネス ニーズに合わせた、焦点を絞った特化したサービスが可能になります。独立してデプロイ可能: マイクロサービスは、独立してデプロイ可能なユニットとして設計されています。各マイクロサービスは、他のサービスに影響を与えることなく、個別に開発、テスト、デプロイできます。これにより、デプロイ プロセスで俊敏性とスケーラビリティが実現します。メンテナンス性が高い:
マイクロサービスは、多くの場合、メンテナンス性が高くなるように設計されます。マイクロサービスは範囲が狭く、特定のタスクに焦点が当てられているため、個々のサービスの管理とメンテナンスが容易になります。さらに、マイクロサービスはシステム全体に影響を及ぼすことなく更新、パッチ適用、または置き換えが可能で、アプリケーションのメンテナンスと進化が容易になります。したがって、マイクロサービスが通常満たさない基準は、密結合です。

最新問題: **20**
組織では、マイクロサービスに使用されるすべてのデプロイ済みコンテナ イメージに、指定されたマスター暗号化キー (MEK) で署名することを義務付けています。ビルド プロセスの一環としてコンテナ イメージに適切に署名しましたが、Oracle Cloud Infrastructure (OCI) Container Engine for Kubemetes (OKE) クラスタにデプロイされるときに、自動的に検証されるようにする必要があります。MEK がすでに利用可能な OCI Vault に保存されて

いると仮定して、OKE クラスタにデプロイするときにイメージの検証を義務付けるには、どのオプションを使用する必要がありますか? (最適な回答を選択してください。)
A. これはポッド レベルで適用されるため、Kubemetes ポッドのデプロイメントごとにイメージ検証ポリシーを個別に有効にします。
B. これはノード プール レベルで適用されるため、各 OKE クラスター内の各ノード プールに対してイメージ検証ポリシーを個別に有効にします。
C. これはクラスター レベルで適用されるため、各 OKE クラスターごとにイメージ検証ポリシーを個別に有効にします。
（正しい）
D. OKE サービス コントロール プレーンのイメージ検証ポリシーを有効にします。これにより、すべての OKE クラスターにこれが適用されます。
Answer: C (メーッセージを残す**)**
コンテナ イメージを Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE) クラスタにデプロイするときにイメージ検証を義務付けるには、各 OKE クラスタに対して個別にイメージ検証ポリシーを有効にする必要があります。これはクラスタ レベルで適用されます。クラスタ レベルでイメージ検証ポリシーを有効にすると、OKE クラスタにデプロイされたすべてのコンテナ イメージが、指定されたマスター暗号化キー (MEK) に対して自動的に検証されます。これにより、署名された信頼できるコンテナ イメージのみが使用されるようになり、デプロイされたマイクロサービスのセキュリティと整合性が維持されます。クラスタ レベルでイメージ検証ポリシーを有効にすると、クラスタ内のすべてのノードとノード プールにわたって検証プロセスを一貫して一元的に適用できます。これにより、クラスタ全体のイメージ検証に対する標準化されたアプローチが提供され、管理が簡素化され、組織の義務への準拠が保証されます。ノード プールごとに個別に、またはポッド レベルでイメージ検証ポリシーを有効にすると、検証プロセスが複雑になり、矛盾が生じる可能性があります。したがって、クラスタ レベルでイメージ検証を適用することが推奨される方法です。

最新問題: **21**
ある会社では、トランザクションをリアルタイムで処理する必要がある新しいアプリケーションを開発しています。会社は、すべてのトランザクションが順番に処理され、トランザクションが失われないようにしたいと考えています。このシナリオで OCI キューを活用するための正しい戦略はどれですか。
A. トランザクションの種類ごとに個別のキューを使用します。
B. 単一のキューを使用してすべてのトランザクションを処理します。
C. アプリケーション インスタンスごとに個別のキューを使用します。
D. 優先キューを使用してリクエストに優先順位を付けます。
Answer: (解答を表示する**)**
OCI キューは、サーバーレス方式で非同期 (分離) 通信を可能にするサービスです3。キューは、損失や重複のない独立した処理を必要とする大量のトランザクション データを処理し

ます3。キューは、FIFO (先入れ先出し) 配信オプション3 を使用して、キュー内のメッセージの順序付けをサポートします。したがって、単一のキューを使用してすべてのトランザクションを処理すると、すべてのトランザクションが順番に処理され、トランザクションが失われることがなくなります。検証済みの参照: キューの概要

最新問題: **22**
DevOps 方法論における継続的デリバリーと継続的デプロイメントの違いは何ですか? （最適な回答を選択してください。）
A. 継続的デリバリーには開発者タスクの自動化が含まれますが、継続的デプロイメントには手動の運用タスクが含まれます。
B. 継続的デリバリーには自動リンティングが必要ですが、継続的デプロイメントのテストは手動で実行する必要があります。
C. 継続的デリバリーでは開発環境への自動デプロイメントが利用されますが、継続的デプロイメントでは運用環境への自動デプロイメントが行われます。
D. 継続的デリバリーは手動でデプロイメントを開始するプロセスですが、継続的デプロイメントはデプロイメント プロセスの自動化に基づいています。
Answer: (解答を表示する**)**
DevOps ライフサイクルにおける継続的デリバリーと継続的デプロイメントの 2 つの正しい違いは次のとおりです。継続的デリバリーは手動でデプロイメントを開始するプロセスですが、継続的デプロイメントはデプロイメント プロセスの自動化に基づいています。継続的デリバリーでは、ソフトウェアはデプロイメントの準備ができていますが、デプロイメントの決定は人間が手動で行います。一方、継続的デプロイメントではデプロイメント プロセスが自動化され、ソフトウェアが必要なすべてのテストと品質チェックに合格すると、人間の介入なしに自動的にデプロイされます。継続的デリバリーでは開発環境への自動デプロイメントを利用しますが、継続的デプロイメントでは実稼働環境への自動デプロイメントが含まれます。継続的デリバリーでは、ソフトウェアは開発環境またはステージング環境に自動的にデプロイされ、さらにテストと検証が行われます。ただし、実稼働環境への実際のデプロイは手動で行われます。継続的デプロイメントでは、ソフトウェアは実稼働環境に自動的にデプロイされるため、デプロイメント プロセスで手動介入する必要がなくなります。これらの違いは、DevOps ライフサイクルにおける継続的デリバリーと継続的デプロイメントのアプローチ間のデプロイメント プロセスにおける自動化と人間の関与のレベルを強調しています。

最新問題: **23**
DevOps プロセスの観点からは、アプリケーションへの変更をバージョン管理下に置くことが推奨されます。次のどれが、Docker イメージへの変更をバージョン管理システムに保存することを可能にしますか?
A. docker-compose.yml を更新しています
B. docker commit を実行しています
C. docker save を実行しています

D. Dockerfile を更新しています
Answer: (解答を表示する**)**
Docker イメージへの変更をバージョン管理システムに保存できるようにするオプションは、docker commit です。docker commit コマンドは、コンテナーの変更から新しいイメージを作成するために使用されます。実行中のコンテナーを入力として受け取り、そのコンテナーに加えられた変更をキャプチャし、それらの変更を含む新しいイメージを作成します。この新しいイメージは、タグ付けしてレジストリにプッシュするか、ローカルに保存できます。docker commit を使用すると、コンテナーに加えられた変更を新しいイメージとして効果的にキャプチャし、Dockerfile やその他のプロジェクト ファイルとともにバージョン管理システムに保存できます。これにより、時間の経過に伴う Docker イメージへの変更の再現性と追跡可能性が実現します。

最新問題**: 24**
Oracle Functions の最大実行時間はどれくらいですか?
A. 240秒
B. 300秒
C. 60秒
D. 120秒
Answer: (解答を表示する**)**
Oracle Functions の最大実行時間は 300 秒で、これは 5 分に相当します。つまり、Oracle Functions 内で実行される関数は、実行時間が 5 分を超えることはできません。関数の実行時間がこれより長い場合は、外部サービスを非同期に呼び出したり、長時間実行されるプロセスを使用したりといった代替アプローチを検討する必要があります。Oracle Functions プラットフォーム内で最適なパフォーマンスと効率を確保するには、この実行時間制限を考慮して関数を設計することが重要です。

最新問題**: 25**
Kubernetes でローリング アップデート デプロイメント戦略を実装するには、どの kubectl コマンド構文が有効ですか? (最適な回答を選択してください。)
A. kubectl アップグレード -c <コンテナ> --image=image:v2
B. kubectl update <デプロイメント名> --image=image:v2
C. kubectl rolling-update <デプロイメント名> --image=image:v2
D. kubectl update -c <コンテナ> --iniage=イメージ: v2
Answer: C (メーッセージを残す**)**
kubectl コマンドを使用して Kubernetes でローリング アップデート デプロイメント戦略を実装するための正しい構文は次のとおりです: kubectl rolling-update <deployment-name> --image=image:v2 このコマンドは、コンテナー イメージを image:v2 に更新することで、指定されたデプロイメントのローリング アップデートを開始します。ローリング アップデート戦略により、可用性を維持し、ダウンタイムを最小限に抑えながら、アプリケーションの新しいバージョンが段階的にデプロイされます。

最新問題: **26**
Oracle Functions は、デプロイされたすべての関数を監視し、さまざまなメトリックを収集してレポートします。Oracle Cloud Infrastructure (OCI) コンソールでアプリケーション メトリックを表示するときに使用できないのはどれですか。
A. 関数が実行される時間の長さ。
B. エラーにより失敗するまでに関数が実行した再試行回数。
C. スロットルにより失敗した関数呼び出し要求の数。
D. エラー応答で失敗した関数呼び出し要求の数。
Answer: (解答を表示する**)**
Oracle Cloud Infrastructure (OCI) コンソールでアプリケーション メトリックを表示するときに使用できないオプションは、「エラーが原因で失敗するまでに関数が行った再試行の回数」です。Oracle Functions の OCI コンソールでアプリケーション メトリックを表示すると、通常、関数のパフォーマンスと使用状況に関連するメトリックが表示されます。これらのメトリックは、関数のパフォーマンスと使用状況に関する洞察を提供します。通常、次のメトリックを使用できます。スロットルにより失敗した関数の呼び出し要求の数: このメトリックは、構成された同時実行制限またはスロットル設定に達したために関数によって処理されなかった要求の数を示します。関数の実行時間: このメトリックは、各関数呼び出しの期間を表し、関数が実行を完了するまでにかかる時間を測定します。エラー応答で失敗した関数の呼び出し要求の数: このメトリックは、関数の呼び出し中にエラーが発生し、応答が失敗した要求の数をカウントします。ただし、エラーにより失敗するまでに関数によって行われた再試行の回数は、通常、OCI コンソールのアプリケーション メトリックの一部としては利用できません。関数によって行われた再試行は通常、呼び出し元レベルで処理され、再試行の具体的な詳細は、アプリケーション レベルのメトリックの一部としてキャプチャされない可能性があります。メトリックの可用性とその具体的な詳細は、Oracle Functions のバージョンと構成、および監視設定によって異なる場合があることに注意してください。利用可能なメトリックに関する正確で最新の情報については、Oracle Functions のドキュメントを参照し、公式ドキュメントを参照することをお勧めします。

最新問題: **27**
Kubernetes アーキテクチャ内のマスター マシンとワーカー マシンによって形成されるグループを表す用語はどれですか。
A. クラスター
B. ノード
C. デプロイメント
D. コンテナ
E. ポッド
Answer: A (メーッセージを残す**)**

Kubernetes アーキテクチャでマスター マシンとワーカー マシンによって形成されるグループを表す用語は、「クラスター」です。Kubernetes のクラスターは、1 台以上のマスター マシンと複数のワーカー マシン (ノードとも呼ばれます) で構成されます。マスター マシンは、コントロール プレーン全体を管理し、ワーカー ノード上のコンテナーの展開と管理を調整します。ワーカー ノードは、コンテナーの実行とワークロードの実行を担当します。クラスターは、Kubernetes の組織と管理の基本単位であり、コンテナー化されたアプリケーションを実行および管理するためのインフラストラクチャとリソースを提供します。クラスターは、その中に展開されたアプリケーションの高可用性、スケーラビリティ、およびフォールト トレランスを保証します。

最新問題**: 28**
(CHK_1>3) 顧客のトランザクション データを Oracle Cloud Infrastructure (OCI) ストリーミング サービスにロードする電子商取引アプリケーションがあります。データを抽出して変換してから、サードパーティの REST エンドポイントに送信する必要があります。このプロセスを自動化するには、OCI サービス コネクタ ハブを活用するように指示されています。この要件に対応する構成オプションはどれですか。
A. 新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: 関数 * ターゲット:
機能
B. 新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: API ゲートウェイ * ターゲット:
通知
C. 新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: なし * ターゲット: 通知
D. 新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: API ゲートウェイ * ターゲット:
機能
E. 新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: 関数 * ターゲット: API ゲートウェイ
Answer: (解答を表示する**)**
Oracle Cloud Infrastructure (OCI)ストリーミングサービスからデータを抽出して変換し、OCIサービスコネクタハブを使用してサードパーティのRESTエンドポイントに送信するという要件に対応するには、次の構成オプションが最適です。新しいサービスコネクタを次のように構成します。* ソース: ストリーミング * タスク: なし
* ターゲット: 通知 ストリーミング サービスをソースとして選択すると、ストリームからトランザクション データを取得できます。データを変換してサードパーティの REST エンドポイントに送信する必要があるため、コネクタで特定のタスクを指定する必要はありません。ターゲットは通知に設定されており、これにより、変換されたデータを OCI 環境外のエンドポイントに送信できます。通知は、HTTP エンドポイント、電子メール アドレスな

ど、サポートされているさまざまな送信先にデータを配信するように構成できます。この構成により、ストリーミング サービスからデータを抽出して目的のサードパーティ REST エンドポイントに送信するプロセスを自動化し、データの抽出、変換、転送の要件を満たすことができます。

最新問題: **29**
ブルー/グリーン デプロイメント戦略を使用する場合、クラウド ネイティブ アプリケーションのバージョンは OKE クラスターにどのようにデプロイされますか?
A. 現在のアプリケーションは、徐々に新しいアプリケーション バージョンに置き換えられます。
B. 新しいアプリケーション バージョンは、選択されたグループのユーザーに少しずつ展開されます。
C. 古いアプリケーション バージョンと新しいアプリケーション バージョンの両方が同時に本番環境にデプロイされます。
Answer: C (メーッセージを残す**)**
ブルー/グリーン デプロイメント戦略では、2 つの同一環境を使用してアプリケーションの新バージョンをリリースできます。そのうちの 1 つは特定の時間にアクティブです。アプリケーションの現在のバージョンはアクティブ環境にプロビジョニングされ、新しいバージョンはスタンバイ環境にデプロイされます1。トラフィックは、イングレス リソースを更新することでアクティブ環境からスタンバイ環境にシフトされます2。したがって、古いアプリケーション バージョンと新しいアプリケーション バージョンの両方が同時に本番環境にデプロイされますが、トラフィックを受信するのはそのうちの 1 つだけです。検証済みの参照: OCI DevOps サービスの新しいデプロイメント戦略の発表、ブルーグリーン OKE デプロイメント

最新問題: **30**
Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE) で実行されるサービスにログ記録を実装する予定です。適切なログ記録アプローチを説明する記述はどれですか。
A. 各サービスは独自のログ ファイルにログを記録します。
B. すべてのサービスは外部ログ システムにログを記録します。
C. すべてのサービスは標準出力にのみログを記録します。
D. すべてのサービスAAは共有ログファイルに記録されます。
Answer: C (メーッセージを残す**)**

最新問題: **31**
運用環境で実行されているマイクロサービス A と B が 2 つあります。サービス A はサービス B の API に依存しています。サービス B を含むすべての依存関係をデプロイせずに、サービス A への変更をテストしたいと考えています。サービス A をテストするには、どのアプローチを採用すればよいでしょうか。

A. サービス B の以前のテスト バージョンを使用してテストします。
B. サービス B の API モックを使用してテストします。
C. サービス B の現在の本番バージョンを使用してテストします。
D. サービス B は依存関係であるため、これは不可能です。
Answer: B (メーッセージを残す**)**
正解は、サービス B の API モックを使用してテストすることです。サービス B を含むすべての依存関係をデプロイせずにサービス A をテストするには、サービス B の API モックを使用できます。API モックは、実際のサービスの動作を模倣した API のシミュレートされたバージョンです。API モックを使用すると、サービス A のテストを分離し、サービス B の API の応答と動作をシミュレートできます。API モックを使用すると、サービス B の API の予想される応答と動作を定義できるため、サービス A とサービス B のモック バージョン間の統合をテストできます。このアプローチにより、実際のサービス B の可用性や変更に依存せずに、サービス A の機能を検証できます。依存関係を切り離し、API モックを使用すると、サービス A の独立したテストを実行し、その機能を分離して確保できます。

最新問題**: 32**
Oracle Functions アプリケーションを正確に説明する 2 つのステートメントはどれですか? (2 つ選択してください。)
A. アプリケーション内のすべての関数で使用できる構成変数を格納するための共通コンテキスト。
同じ構成を共有するすべての関数を含む Docker イメージ。
B. Oracle Functions、Oracle Cloud Infrastructure (OCI) イベント、および OCI API ゲートウェイ サービスに基づくアプリケーション。
C. OCI イベント サービスに応答して呼び出される小さなコード ブロック。
関数の論理グループ。
D. 同じ構成を共有するすべての関数を含む Docker イメージ。
Answer: A,C (メーッセージを残す**)**
正しい記述は次のとおりです。アプリケーション内のすべての関数で使用できる構成変数を格納する共通コンテキスト。同じ構成を共有するすべての関数を含む Docker イメージ。関数の論理グループ。説明: Oracle Functions アプリケーションは、アプリケーション内の関数に共通コンテキストを提供します。これにより、アプリケーション内のすべての関数

からアクセスできる構成変数を格納できます。同じアプリケーション内の関数は、共通の構成と依存関係を含む同じ Docker イメージを共有できます。Oracle Functions アプリケーションは、関連する関数を整理する論理グループとして機能します。同じアプリケーション内の関数はまとめて管理でき、アプリケーション コンテキスト内で相互作用してリソースを共有できます。

最新問題: **33**
Kubernetes クラスターの削除に関する次の 2 つの記述のうち正しいものはどれですか? (2 つ選択してください。)
A. クラスターを削除すると、クラスター作成プロセス中に作成された、またはクラスターに関連付けられた他のリソース (VCNS、インターネット ゲートウェイ、NAT ゲートウェイ、ルート テーブル、セキュリティ リスト、ロード バランサ、ブロック ボリュームなど) が自動的に削除されます。
B. ワーカー ノードの自動生成された名前を変更してからクラスターを削除しても、名前が変更されたワーカー ノードは削除されません。
C. クラスターを削除しても、クラスター作成プロセス中に作成されたリソースやクラスターに関連付けられたリソース (VCNS、インターネット ゲートウェイ、NAT ゲートウェイ、ルート テーブル、セキュリティ リスト、ロード バランサ、ブロック ボリュームなど) は自動的に削除されません。
D. ワーカー ノードの自動生成された名前を変更しても、ワーカー ノードが作成されたクラスターが削除されたときにワーカー ノードが削除されることには影響しません。
E. Kubernetes クラスター内の oke-c<part-of cluster- CCID>-<part-of-node-pool-OCID>-<part-of-subnet-OCID>-<slot> 形式のワーカー ノードの自動生成された名前を変更することはできません。
Answer: B,C (メーッセージを残す**)**
Kubernetes クラスターの削除に関する正しい記述は次のとおりです。ワーカー ノードの自動生成された名前を変更してからクラスターを削除した場合、名前が変更されたワーカー ノードは削除されません。クラスターを削除すると、ワーカー ノードの名前を変更しても、その削除には影響しません。クラスターの削除プロセスでは、名前が変更されたワーカー ノードは考慮されず、クラスターに関連付けられているすべてのワーカー ノードが削除されます。クラスターを削除しても、クラスター作成プロセス中に作成された、またはクラスターに関連付けられている他のリソース (VCN、インターネット ゲートウェイ、NAT ゲートウェイ、ルート テーブル、セキュリティ リスト、ロード バランサ、ブロック ボリュームなど) は自動的に削除されません。これらの追加リソースは、クラスターを削除しても自動的に削除されません。必要に応じて、これらのリソースの削除を個別に管理する必要があります。したがって、クラスターを削除しても名前が変更されたワーカー ノードは削除されず、クラスターを削除しても他の関連リソースは自動的に削除されないというのが正しい記述です。

最新問題: **34**
Oracle Cloud Infrastructure (OCI) Resource Manager を使用してインフラストラクチャのライフサイクルを管理しており、Terraform アクションが開始するたびに電子メールを受信したいとします。コードを記述せずにこれを実現するには、OCI イベント サービスをどのように使用すればよいでしょうか。
A. OCI イベント サービスで、『リソース マネージャー スタック - 更新』条件に一致するルールを作成します。次に、『アクション タイプ: 電子メール』を選択し、送信先の電子メール アドレスを指定します。
B. 宛先の電子メール アドレスを使用して、OCI 通知トピックと電子メール サブスクリプションを作成します。次に、『リソース マネージャ ジョブ - 作成』条件に一致する OCI イベント ルールを作成し、対応するアクションの通知トピックを選択します。
C. 宛先の電子メール アドレスを使用して OCI 電子メール配信構成を作成します。次に、『リソース マネージャ ジョブ - 作成』条件に一致する OCI イベント ルールを作成し、対応するアクションの電子メール構成を選択します。
D. 宛先の電子メール アドレスを使用して、OCI 通知トピックと電子メール サブスクリプションを作成します。次に、『リソース マネージャ スタック - 更新』条件に一致する OCI イベント ルールを作成し、対応するアクションの通知トピックを選択します。
Answer: B (メーッセージを残す**)**
Oracle Cloud Infrastructure (OCI) Resource Manager で Terraform アクションが開始されるたびに、コードを記述せずに電子メールを受信するための正しい方法は次のとおりです。送信先電子メール アドレスを使用して、OCI 通知トピックと電子メール サブスクリプションを作成します。これにより、電子メール配信構成が定義されます。『リソース マネージャ ジョブ - 作成』条件に一致する OCI イベント ルールを作成します。このルールは、リソース マネージャ ジョブの作成時にトリガーされます。OCI イベント ルールで、手順 1 で作成した通知トピックを、対応するイベントのアクションとして選択します。これにより、指定した電子メール アドレスに通知が送信されるようになります。これらの手順に従うことで、OCI Resource Manager でリソース マネージャ ジョブが作成されるたびに電子メール通知を送信するように OCI イベント サービスを構成できます。