



OpenShift Container Platform 4.5

Azure へのインストール

OpenShift Container Platform Azure クラスターのインストール

OpenShift Container Platform 4.5 Azure へのインストール

OpenShift Container Platform Azure クラスターのインストール

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installing_on_Azure.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書では、Microsoft Azure に OpenShift Container Platform クラスターをインストールし、アンインストールする方法について説明します。

目次

第1章 AZURE へのインストール	6
1.1. AZURE アカウントの設定	6
1.1.1. Azure アカウントの制限	6
1.1.2. Azure でのパブリック DNS ゾーンの設定	9
1.1.3. Azure アカウント制限の拡張	9
1.1.4. 必要な Azure ロール	10
1.1.5. サービスプリンシパルの作成	10
1.1.6. サポート対象の Azure リージョン	13
1.1.7. 次のステップ	14
1.2. クラスターの AZURE へのクイックインストール	15
1.2.1. 前提条件	15
1.2.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	15
1.2.3. SSH プライベートキーの生成およびエージェントへの追加	16
1.2.4. インストールプログラムの取得	17
1.2.5. クラスターのデプロイ	18
1.2.6. バイナリーのダウンロードによる CLI のインストール	20
1.2.6.1. Linux への CLI のインストール	20
1.2.6.2. Windows での CLI のインストール	21
1.2.6.3. macOS への CLI のインストール	21
1.2.7. クラスターへのログイン	22
1.2.8. 次のステップ	22
1.3. カスタマイズによる AZURE へのクラスターのインストール	22
1.3.1. 前提条件	22
1.3.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	23
1.3.3. SSH プライベートキーの生成およびエージェントへの追加	23
1.3.4. インストールプログラムの取得	25
1.3.5. インストール設定ファイルの作成	25
1.3.5.1. インストール設定パラメーター	27
1.3.5.1.1. 必須設定パラメーター	27
1.3.5.1.2. ネットワーク設定パラメーター	29
1.3.5.1.3. オプションの設定パラメーター	30
1.3.5.1.4. 追加の Azure 設定パラメーター	34
1.3.5.2. Azure のカスタマイズされた install-config.yaml ファイルのサンプル	35
1.3.6. クラスターのデプロイ	37
1.3.7. バイナリーのダウンロードによる CLI のインストール	38
1.3.7.1. Linux への CLI のインストール	38
1.3.7.2. Windows での CLI のインストール	39
1.3.7.3. macOS への CLI のインストール	39
1.3.8. クラスターへのログイン	40
1.3.9. 次のステップ	40
1.4. ネットワークのカスタマイズによる AZURE へのクラスターのインストール	40
1.4.1. 前提条件	40
1.4.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	41
1.4.3. SSH プライベートキーの生成およびエージェントへの追加	41
1.4.4. インストールプログラムの取得	43
1.4.5. インストール設定ファイルの作成	43
1.4.5.1. インストール設定パラメーター	45
1.4.5.1.1. 必須設定パラメーター	45
1.4.5.1.2. ネットワーク設定パラメーター	47
1.4.5.1.3. オプションの設定パラメーター	48
1.4.5.1.4. 追加の Azure 設定パラメーター	52

1.4.5.2. ネットワーク設定パラメーター	53
1.4.5.3. Azure のカスタマイズされた install-config.yaml ファイルのサンプル	54
1.4.6. 高度なネットワーク設定パラメーターの変更	56
1.4.7. Cluster Network Operator (CNO) の設定	57
1.4.7.1. OpenShift SDN デフォルト CNI ネットワークプロバイダーの設定パラメーター	58
1.4.7.2. OVN-Kubernetes デフォルト CNI ネットワークプロバイダーの設定パラメーター	59
1.4.7.3. Cluster Network Operator の設定例	60
1.4.8. クラスターのデプロイ	60
1.4.9. バイナリーのダウンロードによる CLI のインストール	61
1.4.9.1. Linux への CLI のインストール	62
1.4.9.2. Windows での CLI のインストール	62
1.4.9.3. macOS への CLI のインストール	63
1.4.10. クラスターへのログイン	63
1.4.11. 次のステップ	64
1.5. AZURE のクラスターの既存 VNET へのインストール	64
1.5.1. 前提条件	64
1.5.2. OpenShift Container Platform クラスターでの VNet の再利用について	64
1.5.2.1. VNet を使用するための要件	64
1.5.2.1.1. ネットワークセキュリティグループの要件	65
1.5.2.2. パーMISSIONの区分	66
1.5.2.3. クラスター間の分離	66
1.5.3. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	67
1.5.4. SSH プライベートキーの生成およびエージェントへの追加	67
1.5.5. インストールプログラムの取得	68
1.5.6. インストール設定ファイルの作成	69
1.5.6.1. インストール設定パラメーター	71
1.5.6.1.1. 必須設定パラメーター	71
1.5.6.1.2. ネットワーク設定パラメーター	73
1.5.6.1.3. オプションの設定パラメーター	74
1.5.6.1.4. 追加の Azure 設定パラメーター	78
1.5.6.2. Azure のカスタマイズされた install-config.yaml ファイルのサンプル	79
1.5.6.3. インストール時のクラスター全体のプロキシの設定	81
1.5.7. クラスターのデプロイ	83
1.5.8. バイナリーのダウンロードによる CLI のインストール	84
1.5.8.1. Linux への CLI のインストール	84
1.5.8.2. Windows での CLI のインストール	84
1.5.8.3. macOS への CLI のインストール	85
1.5.9. クラスターへのログイン	85
1.5.10. 次のステップ	86
1.6. プライベートクラスターの AZURE へのインストール	86
1.6.1. 前提条件	86
1.6.2. プライベートクラスター	86
1.6.2.1. Azure のプライベートクラスター	87
1.6.2.1.1. 制限事項	87
1.6.3. OpenShift Container Platform クラスターでの VNet の再利用について	87
1.6.3.1. VNet を使用するための要件	88
1.6.3.1.1. ネットワークセキュリティグループの要件	89
1.6.3.2. パーMISSIONの区分	89
1.6.3.3. クラスター間の分離	90
1.6.4. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	90
1.6.5. SSH プライベートキーの生成およびエージェントへの追加	90
1.6.6. インストールプログラムの取得	92
1.6.7. インストール設定ファイルの手動作成	93

1.6.7.1. インストール設定パラメーター	93
1.6.7.1.1. 必須設定パラメーター	94
1.6.7.1.2. ネットワーク設定パラメーター	95
1.6.7.1.3. オプションの設定パラメーター	97
1.6.7.1.4. 追加の Azure 設定パラメーター	100
1.6.7.2. Azure のカスタマイズされた install-config.yaml ファイルのサンプル	101
1.6.7.3. インストール時のクラスター全体のプロキシの設定	103
1.6.8. クラスターのデプロイ	105
1.6.9. バイナリーのダウンロードによる CLI のインストール	106
1.6.9.1. Linux への CLI のインストール	106
1.6.9.2. Windows での CLI のインストール	107
1.6.9.3. macOS への CLI のインストール	107
1.6.10. クラスターへのログイン	108
1.6.11. 次のステップ	108
1.7. ARM テンプレートを使用したクラスターの AZURE へのインストール	108
1.7.1. 前提条件	109
1.7.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス	109
1.7.3. Azure プロジェクトの設定	110
1.7.3.1. Azure アカウントの制限	110
1.7.3.2. Azure でのパブリック DNS ゾーンの設定	113
1.7.3.3. Azure アカウント制限の拡張	113
1.7.3.4. 証明書署名要求の管理	114
1.7.3.5. 必要な Azure ロール	114
1.7.3.6. サービスプリンシパルの作成	114
1.7.3.7. サポート対象の Azure リージョン	117
1.7.4. インストールプログラムの取得	118
1.7.5. SSH プライベートキーの生成およびエージェントへの追加	119
1.7.6. AWS のインストールファイルの作成	121
1.7.6.1. インストール設定ファイルの作成	121
1.7.6.2. インストール時のクラスター全体のプロキシの設定	122
1.7.6.3. ARM テンプレートの一般的な変数のエクスポート	124
1.7.6.4. Kubernetes マニフェストおよび Ignition 設定ファイルの作成	125
1.7.7. Azure リソースグループおよびアイデンティティの作成	128
1.7.8. RHCOS クラスターイメージおよびブートストラップ Ignition 設定ファイルのアップロード	128
1.7.9. DNS ゾーンの作成例	130
1.7.10. Azure での VNet の作成	131
1.7.10.1. VNet の ARM テンプレート	131
1.7.11. Azure インフラストラクチャー用の RHCOS クラスターイメージのデプロイ	133
1.7.11.1. イメージストレージの ARM テンプレート	134
1.7.12. ユーザーによってプロビジョニングされるインフラストラクチャーのネットワーク要件	135
ネットワークポロジータン	136
ロードバランサー	136
1.7.13. Azure でのネットワークおよび負荷分散コンポーネントの作成	138
1.7.13.1. ネットワークおよびロードバランサーの ARM テンプレート	139
1.7.14. Azure でのブートストラップマシンの作成	144
1.7.14.1. ブートストラップマシンの ARM テンプレート	145
1.7.15. Azure でのコントロールプレーンの作成	149
1.7.15.1. コントロールプレーンマシンの ARM テンプレート	151
1.7.16. ブートストラップの完了を待機し、Azure のブートストラップリソースを削除する	156
1.7.17. Azure での追加のワーカーマシンの作成	157
1.7.17.1. ワーカーマシンの ARM テンプレート	158
1.7.18. バイナリーのダウンロードによる CLI のインストール	163
1.7.18.1. Linux への CLI のインストール	163

1.7.18.2. Windows での CLI のインストール	163
1.7.18.3. macOS への CLI のインストール	164
1.7.19. クラスターへのログイン	164
1.7.20. マシンの証明書署名要求の承認	165
1.7.21. Ingress DNS レコードの追加	167
1.7.22. ユーザーによってプロビジョニングされるインフラストラクチャーでの Azure インストールの実行	169
1.8. AZURE でのクラスターのアンインストール	169
1.8.1. インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターの削除	169

第1章 AZURE へのインストール

1.1. AZURE アカウントの設定

OpenShift Container Platform をインストールする前に、Microsoft Azure アカウントを設定する必要があります。



重要

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

1.1.1. Azure アカウントの制限

OpenShift Container Platform クラスターは数多くの Microsoft Azure コンポーネントを使用し、デフォルトの [Azure サブスクリプションおよびサービス制限、クォータ、および制約](#) は、OpenShift Container Platform クラスターをインストールする機能に影響を与えます。



重要

デフォルトの制限は、Free Trial や Pay-As-You-Go、および DV2、F、および G などのシリーズといったカテゴリータイプによって異なります。たとえば、Enterprise Agreement サブスクリプションのデフォルトは 350 コアです。

サブスクリプションタイプの制限を確認し、必要に応じて、デフォルトのクラスターを Azure にインストールする前にアカウントのクォータ制限を引き上げます。

以下の表は、OpenShift Container Platform クラスターのインストールおよび実行機能に影響を与える可能性のある Azure コンポーネントの制限を要約しています。

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明
---------	--------------------	-----------------	----

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明
vCPU	40	リージョンごとに 20	<p>デフォルトのクラスターには 40 の vCPU が必要であるため、アカウントの上限を引き上げる必要があります。</p> <p>デフォルトで、各クラスターは以下のインスタンスを作成します。</p> <ul style="list-style-type: none"> ● 1つのブートストラップマシン。これはインストール後に削除されます。 ● 3つのコントロールプレーンマシン ● 3つのコンピュートマシン <p>ブートストラップマシンは 4 vCPUS を使用する Standard_D4s_v3 マシンを使用し、コントロールプレーンマシンは 8 vCPU を使用する Standard_D8s_v3 仮想マシンを使用し、さらにワーカーマシンは、4 vCPU を使用する Standard_D4s_v3 仮想マシンを使用するため、デフォルトクラスターには 40 の vCPU が必要になります。4 vCPU を使用するブートストラップノードの仮想マシンは、インストール時にのみ使用されます。</p> <p>追加のワーカーノードをデプロイし、自動スケーリングを有効にし、大規模なワークロードをデプロイするか、または異なるインスタンスタイプを使用するには、アカウントの vCPU 制限をさらに引き上げ、クラスターが必要なマシンをデプロイできるようにする必要があります。</p> <p>デフォルトで、インストールプログラムはコントロールプレーンおよびコンピュートマシンを、リージョン内のすべてのアベイラビリティゾーン に分散します。クラスターの高可用性を確保するには、少なくとも 3 つ以上のアベイラビリティゾーンのあるリージョンを選択します。リージョンに含まれるアベイラビリティゾーンが 3 つ未満の場合、インストールプログラムは複数のコントロールプレーンマシンを利用可能なゾーンに配置します。</p>
VNet	1	リージョンごとに 1000	各デフォルトクラスターには、2 つのサブネットを含む 1 つの Virtual Network (VNet) が必要です。
ネットワークインターフェイス	6	リージョンごとに 65,536	各デフォルトクラスターには、6 つのネットワークインターフェイスが必要です。さらに多くのマシンを作成したり、デプロイしたワークロードでロードバランサーを作成する場合、クラスターは追加のネットワークインターフェイスを使用します。

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明						
ネットワークセキュリティグループ	2	5000	<div>各デフォルトクラスター。各クラスターは VNet の各サブネットにネットワークセキュリティグループを作成します。デフォルトのクラスターは、コントロールプレーンおよびコンピューターノードのサブネットにネットワークセキュリティグループを作成します。</div> <table><tr><td>control plane</td><td>任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。</td></tr><tr><td>node</td><td>インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。</td></tr></table>	control plane	任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。	node	インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。		
control plane	任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。								
node	インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。								
ネットワークロードバランサー	3	リージョンごとに 1000	<div>各クラスターは以下の ロードバランサー を作成します。</div> <table><tr><td>default</td><td>ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス</td></tr><tr><td>internal</td><td>コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス</td></tr><tr><td>external</td><td>コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス</td></tr></table> <div>アプリケーションが追加の Kubernetes LoadBalancer サービスオブジェクトを作成すると、クラスターは追加のロードバランサーを使用します。</div>	default	ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス	internal	コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス	external	コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス
default	ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス								
internal	コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス								
external	コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス								
パブリック IP アドレス	3		2 つのパブリックロードバランサーのそれぞれはパブリック IP アドレスを使用します。ブートストラップマシンは、インストール時のトラブルシューティングのためにマシンに SSH を実行できるようにパブリック IP アドレスも使用します。ブートストラップノードの IP アドレスは、インストール時にのみ使用されます。						

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明
プライベート IP アドレス	7		内部ロードバランサー、3つのコントロールプレーンマシンのそれぞれ、および3つのワーカーマシンのそれぞれはプライベート IP アドレスを使用します。

1.1.2. Azure でのパブリック DNS ゾーンの設定

OpenShift Container Platform をインストールするには、使用する Microsoft Azure アカウントに、専用のパブリックホスト DNS ゾーンが必要になります。このゾーンはドメインに対する権威を持っている必要があります。このサービスは、クラスターへの外部接続のためのクラスター DNS 解決および名前検索を提供します。

手順

1. ドメイン、またはサブドメイン、およびレジストラを特定します。既存のドメインおよびレジストラを移行するか、Azure または別のソースから新規のものを取得できます。



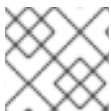
注記

Azure 経由でドメインを購入する方法についての詳細は、Azure ドキュメントの [Buy a custom domain name for Azure App Service](#) を参照してください。

2. 既存のドメインおよびレジストラを使用している場合、その DNS を Azure に移行します。Azure ドキュメントの [Migrate an active DNS name to Azure App Service](#) を参照してください。
3. ドメインの DNS を設定します。Azure ドキュメントの [Tutorial: Host your domain in Azure DNS](#) の手順に従い、ドメインまたはサブドメインのパブリックホストゾーンを作成し、新規の権威ネームサーバーを抽出し、ドメインが使用するネームサーバーのレジストラレコードを更新します。
openshiftcorp.com などのルートドメインや、**clusters.openshiftcorp.com** などのサブドメインを使用します。
4. サブドメインを使用する場合は、所属する会社の手順に従ってその委任レコードを親ドメインに追加します。

1.1.3. Azure アカウント制限の拡張

アカウントの制限を引き上げるには、Azure ポータルでサポートをリクエストします。



注記

サポートリクエストごとに1つの種類のクォータのみを増やすことができます。

手順

1. Azure ポータルの左端で **Help + support** をクリックします。
2. **New support request** をクリックしてから必要な値を選択します。

- a. **Issue type** 一覧から、**Service and subscription limits (quotas)** を選択します。
 - b. **Subscription** 一覧から、変更するサブスクリプションを選択します。
 - c. **Quota type** 一覧から、引き上げるクォータを選択します。たとえば、**Compute-VM (cores-vCPUs) subscription limit increases** を選択し、クラスターのインストールに必要な vCPU の数を増やします。
 - d. **Next: Solutions** をクリックします。
3. **Problem Details** ページで、クォータの引き上げについての必要な情報を指定します。
 - a. **Provide details** をクリックし、**Quota details** ウィンドウに必要な詳細情報を指定します。
 - b. **SUPPORT METHOD** and **CONTACT INFO** セクションに、問題の重大度および問い合わせ先の詳細を指定します。
 4. **Next: Review + create** をクリックしてから **Create** をクリックします。

1.1.4. 必要な Azure ロール

Microsoft Azure アカウントには、使用するサブスクリプションについて以下のロールが必要です。

- **User Access Administrator**

Azure ポータルでロールを設定するには、Azure ドキュメントの [Manage access to Azure resources using RBAC and the Azure portal](#) を参照します。

1.1.5. サービスプリンシパルの作成

OpenShift Container Platform およびそのインストールプログラムは Azure Resource Manager 経由で Microsoft Azure リソースを作成する必要があるため、これを表すサービスプリンシパルを作成する必要があります。

前提条件

- [Azure CLI](#) のインストールまたは更新を実行します。
- **jq** パッケージをインストールします。
- Azure アカウントには、使用するサブスクリプションに必要なロールがなければなりません。

手順

1. Azure CLI にログインします。

```
$ az login
```

認証情報を使用して Web コンソールで Azure にログインします。

2. Azure アカウントでサブスクリプションを使用している場合は、適切なサブスクリプションを使用していることを確認してください。
 - a. 利用可能なアカウントの一覧を表示し、クラスターに使用するサブスクリプションの **tenantId** の値を記録します。

```
$ az account list --refresh
```

出力例

```
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

- b. アクティブなアカウントの詳細を表示し、**tenantId** 値が使用するサブスクリプションと一致することを確認します。

```
$ az account show
```

出力例

```
{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", ❶
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

- ❶ **tenantId** パラメーターの値が適切なサブスクリプションの UUID であることを確認します。

- c. 適切なサブスクリプションを使用していない場合には、アクティブなサブスクリプションを変更します。

```
$ az account set -s <id> ❶
```

- ❶ 使用する必要のあるサブスクリプションの **id** の値を **<id>** の代わりに使用します。

- d. アクティブなサブスクリプションを変更したら、アカウント情報を再度表示します。

```
$ az account show
```

出力例

```
{
  "environmentName": "AzureCloud",
  "id": "33212d16-bdf6-45cb-b038-f6565b61edda",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

- 直前の出力の **tenantId** および **id** パラメーターの値を記録します。OpenShift Container Platform のインストール時にこれらの値が必要になります。
- アカウントのサービスプリンシパルを作成します。

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> ❶
```

- ❶ **<service_principal>** を、サービスプリンシパルに割り当てる名前に置き換えます。

出力例

```
Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}
```

- 直前の出力の **appId** および **password** パラメーターの値を記録します。OpenShift Container Platform のインストール時にこれらの値が必要になります。
- サービスプリンシパルに追加パーミッションを付与します。サービスプリンシパルには、クラスターでそのコンポーネントの認証情報を割り当てられるようにレガシーの **Azure Active Directory Graph** → **Application.ReadWrite.OwnedBy** パーミッションおよび **User Access Administrator** ロールが必要です。
 - User Access Administrator** ロールを割り当てるには、以下のコマンドを実行します。

```
$ az role assignment create --role "User Access Administrator" \
  --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \
    | jq '.[0].objectId' -r) ❶
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

- b. **Azure Active Directory Graph** パーミッションを割り当てるには、以下のコマンドを実行します。

```
$ az ad app permission add --id <appId> \ ❶
  --api 00000002-0000-0000-c000-000000000000 \
  --api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

出力例

```
Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --api
00000002-0000-0000-c000-000000000000" is needed to make the change effective
```

このコマンドで付与する特定のパーミッションについての詳細は、[GUID Table for Windows Azure Active Directory Permissions](#) を参照してください。

- c. パーミッション要求を承認します。アカウントに Azure Active Directory テナント管理者ロールがない場合は、所属する組織のガイドラインに従い、テナント管理者にパーミッション要求を承認するようにリクエストしてください。

```
$ az ad app permission grant --id <appId> \ ❶
  --api 00000002-0000-0000-c000-000000000000
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

1.1.6. サポート対象の Azure リージョン

インストールプログラムは、サブスクリプションに基づいて利用可能な Microsoft Azure リージョンの一覧を動的に生成します。以下の Azure リージョンは OpenShift Container Platform バージョン 4.5.4 でテストされ、検証されています。

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)

- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)
- **eastus2** (East US 2)
- **francecentral** (France Central)
- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)
- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)
- **norwayeast** (Norway East)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

1.1.7. 次のステップ

- OpenShift Container Platform クラスターを Azure にインストールします。[カスタマイズされたクラスターのインストール](#)、またはデフォルトのオプションで [クラスターのクイックインストール](#) を実行できます。

1.2. クラスターの AZURE へのクイックインストール

OpenShift Container Platform バージョン 4.5 では、デフォルトの設定オプションを使用してクラスターを Microsoft Azure にインストールできます。

1.2.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- クラスターをホストできるように [Azure アカウントを設定](#) し、クラスターをデプロイするテスト済みの検証されたリージョンを判別します。
- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。

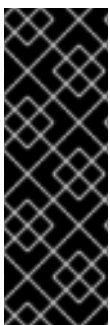
1.2.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

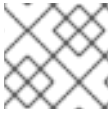


重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.2.3. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N ""
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が `~/.ssh` ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。



注記

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ❶
```

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ❶ `~/.ssh/id_rsa` などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.2.4. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

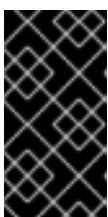
手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

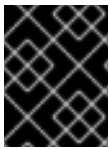
3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.2.5. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に 1 回だけ実行できます。

前提条件

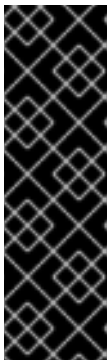
- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

```
$ ./openshift-install create cluster --dir=<installation_directory> \ 1  
--log-level=info 2
```

- 1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。
- 2 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



重要

空のディレクトリーを指定します。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

プロンプト時に値を指定します。

- a. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

- b. ターゲットに設定するプラットフォームとして **azure** を選択します。
- c. お使いのコンピューターに Microsoft Azure プロファイルが保存されていない場合は、サブスクリプションとサービスプリンシパルに以下の Azure パラメーター値を指定します。
 - **azure subscription id** クラスターに使用するサブスクリプション ID。アカウント出力に **id** 値を指定します。
 - **azure tenant id** テナント ID。アカウント出力に **tenantId** 値を指定します。
 - **azure service principal client id** サービスプリンシパルの **appId** パラメーターの値。
 - **azure service principal client secret** サービスプリンシパルの **password** パラメーターの値。
- d. クラスターをデプロイするリージョンを選択します。
- e. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成した Azure DNS ゾーンに対応します。
- f. クラスターの記述名を入力します。



重要

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

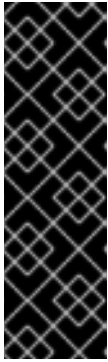
- g. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから取得したプルシークレットを貼り付けます。



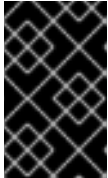
注記

ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。

**重要**

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

**重要**

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

1.2.6. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。

**重要**

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.2.6.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。**PATH** を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.2.6.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.2.6.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.2.7. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。
- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig ❶
```

- ❶ **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

1.2.8. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

1.3. カスタマイズによる AZURE へのクラスターのインストール

OpenShift Container Platform バージョン 4.5 では、インストールプログラムが Microsoft Azure にプロビジョニングするインフラストラクチャーにカスタマイズされたクラスターをインストールできます。インストールをカスタマイズするには、クラスターをインストールする前に、**install-config.yaml** ファイルでパラメーターを変更します。

1.3.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- クラスターをホストできるように [Azure アカウントを設定](#) し、クラスターをデプロイするテスト済みの検証されたリージョンを判別します。

- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。

1.3.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.3.3. SSH プライベートキーの生成およびエージェントへの追加

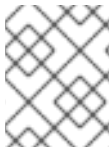
クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が `~/.ssh` ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。



注記

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ①
```

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.3.4. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

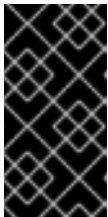
手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.3.5. インストール設定ファイルの作成

Microsoft Azure にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. **install-config.yaml** ファイルを作成します。

- a. 以下のコマンドを実行します。

```
$ ./openshift-install create install-config --dir=<installation_directory> 1
```

- 1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。



重要

空のディレクトリーを指定します。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

- b. プロンプト時に、クラウドの設定の詳細情報を指定します。

- i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

- ii. ターゲットに設定するプラットフォームとして **azure** を選択します。
- iii. お使いのコンピューターに Microsoft Azure プロファイルが保存されていない場合は、サブスクリプションとサービスプリンシパルに以下の Azure パラメーター値を指定します。
 - **azure subscription id** クラスターに使用するサブスクリプション ID。アカウント出力に **id** 値を指定します。
 - **azure tenant id** テナント ID。アカウント出力に **tenantId** 値を指定します。
 - **azure service principal client id** サービスプリンシパルの **appId** パラメーターの値。
 - **azure service principal client secret** サービスプリンシパルの **password** パラメーターの値。

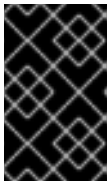
- iv. クラスターをデプロイするリージョンを選択します。
- v. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成した Azure DNS ゾーンに対応します。
- vi. クラスターの記述名を入力します。



重要

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

- vii. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから取得したプルシークレットを貼り付けます。
2. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細については、[インストール設定パラメーターセクション](#)を参照してください。
 3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。



重要

install-config.yaml ファイルはインストールプロセス時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

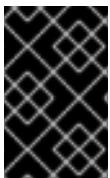
1.3.5.1. インストール設定パラメーター

OpenShift Container Platform クラスターをデプロイする前に、クラスターをホストするクラウドプラットフォームでアカウントを記述し、クラスターのプラットフォームをオプションでカスタマイズするためにパラメーターの値を指定します。**install-config.yaml** インストール設定ファイルを作成する際に、コマンドラインで必要なパラメーターの値を指定します。クラスターをカスタマイズする場合、**install-config.yaml** ファイルを変更して、プラットフォームについての詳細情報を指定できます。



注記

インストール後は、これらのパラメーターを **install-config.yaml** ファイルで変更することはできません。



重要

openshift-install コマンドは、パラメーターのフィールド名を検証しません。正しくない名前を指定すると、関連するファイルまたはオブジェクトは作成されず、エラーが報告されません。指定されたパラメーターのフィールド名が正しいことを確認します。

1.3.5.1.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表1.1 必須パラメーター

パラメーター	説明	値
apiVersion	install-config.yaml コンテンツの API バージョン。現在のバージョンは v1 です。インストーラーは、古い API バージョンをサポートすることもできます。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、 baseDomain と <metadata.name> 。 <baseDomain> 形式を使用する metadata.name パラメーターの値の組み合わせです。	example.com などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース ObjectMeta 。ここからは name パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて {{.metadata.name}} 。 {{.baseDomain}} のサブドメインです。	dev などの小文字、ハイフン (-)、およびピリオド (.) が含まれる文字列。
platform	インストールの実行に使用する特定プラットフォームの設定: aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。 platform.<platform> パラメーターに関する追加情報は、以下の表で特定のプラットフォームについて参照してください。	オブジェクト

パラメーター	説明	値
pullSecret	https://cloud.redhat.com/openshift/install/pull-secret からプルシークレットを取得し、Quay.io などのサービスから OpenShift Container Platform コンポーネントのコンテナイメージのダウンロードを認証します。	<pre>{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }</pre>

1.3.5.1.2. ネットワーク設定パラメーター

既存のネットワークインフラストラクチャーの要件に基づいて、インストール設定をカスタマイズできます。たとえば、クラスターネットワークの IP アドレスブロックを拡張するか、デフォルトとは異なる IP アドレスブロックを指定できます。

IPv4 アドレスのみがサポートされます。

表1.2 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	<p>オブジェクト</p>  <p>注記</p> <p>インストール後に networking オブジェクトで指定したパラメーターを変更することはできません。</p>
networking.networkType	インストールするクラスターネットワークプロバイダー Container Network Interface (CNI) プラグイン。	OpenShiftSDN または OVNKubernetes のいずれか。デフォルト値は OpenShiftSDN です。
networking.clusterNetwork	<p>Pod の IP アドレスブロック。</p> <p>デフォルト値は 10.128.0.0/14 で、ホストの接頭辞は /23 です。</p> <p>複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。</p>	<p>オブジェクトの配列。以下に例を示します。</p> <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>

パラメーター	説明	値
networking.clusterNetwork.cidr	<p>networking.clusterNetwork を使用する場合に必須です。IP アドレスブロック。</p> <p>IPv4 ネットワーク</p>	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は 0 から 32 の間になります。
networking.clusterNetwork.hostPrefix	<p>それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます。hostPrefix 値の 23 は、$2^{(32-23)} - 2$ Pod IP アドレスを提供します。</p>	<p>サブネット接頭辞。</p> <p>デフォルト値は 23 です。</p>
networking.serviceNetwork	<p>サービスの IP アドレスブロック。デフォルト値は 172.30.0.0/16 です。</p> <p>OpenShift SDN および OVN-Kubernetes ネットワークプロバイダーは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。</p>	<p>CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。</p> <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
networking.machineNetwork	<p>マシンの IP アドレスブロック。</p> <p>複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。</p>	<p>オブジェクトの配列。以下に例を示します。</p> <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
networking.machineNetwork.cidr	<p>networking.machineNetwork を使用する場合に必須です。IP アドレスブロック。libvirt 以外のすべてのプラットフォームでは、デフォルト値は 10.0.0.0/16 です。libvirt の場合、デフォルト値は 192.168.126.0/24 です。</p>	<p>CIDR 表記の IP ネットワークブロック。</p> <p>例: 10.0.0.0/16</p> <div>  <div> <p>注記</p> <p>優先される NIC が置かれている CIDR に一致する networking.machineNetwork を設定します。</p> </div> </div>


1.3.5.1.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表1.3 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定される際にも使用できます。	文字列
compute	コンピュートノードを設定するマシンの設定。	machine-pool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
compute.hyperthreading	<p>コンピュートマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div data-bbox="491 1216 600 1503" data-label="Image"> </div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p>	Enabled または Disabled
compute.name	compute を使用する場合に必須です。マシンプールの名前。	worker
compute.platform	compute を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は controlPlane.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	2 以上の正の整数。デフォルト値は 3 です。

パラメーター	説明	値
controlPlane	コントロールプレーンを設定するマシンの設定。	MachinePool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
controlPlane.hyperthreading	<p>コントロールプレーンマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div>  <div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p> </div> </div>	Enabled または Disabled
controlPlane.name	controlPlane を使用する場合に必須です。マシンプールの名前。	master
controlPlane.platform	controlPlane を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は compute.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は 3 のみです (これはデフォルト値です)。

パラメーター	説明	値
fips	<p>FIPS モードを有効または無効にします。デフォルトは false (無効) です。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。</p> <div>  <div> 注記 <p>Azure File ストレージを使用している場合、FIPS モードを有効にすることはできません。</p> </div> </div>	false または true
imageContentSources	release-image コンテンツのソースおよびリポジトリ。	オブジェクトの配列。この表の以下の行で説明されているように、 source およびオプションで mirrors が含まれます。
imageContentSources.source	imageContentSources を使用する場合に必須です。ユーザーが参照するリポジトリを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリを1つ以上指定します。	文字列の配列。
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	Internal または External 。プライベートクラスターをデプロイするには、 publish を Internal に設定します。これはインターネットからアクセスできません。デフォルト値は External です。

パラメーター	説明	値
sshKey	<p>クラスターマシンへのアクセスを認証するための SSH キー。</p> <div>  <p>注記</p> <p>インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、ssh-agent プロセスが使用する SSH キーを指定します。</p> </div>	たとえば、 sshKey: ssh-ed25519 AAAA.. です。

1.3.5.1.4. 追加の Azure 設定パラメーター

追加の Azure 設定パラメーターは以下の表で説明されています。

表1.4 追加の Azure パラメーター

パラメーター	説明	値
controlPlane.platform.azure.osDisk.diskSizeGB	VM の Azure ディスクのサイズ。	GB 単位でディスクのサイズを表す整数。サポートされる最小のディスクサイズは 1024 です。
platform.azure.baseDomainResourceGroupName	ベースドメインの DNS ゾーンが含まれるリソースグループの名前。	文字列 (例: production_cluster)。
platform.azure.region	クラスターをホストする Azure リージョンの名前。	centralus などの有効なリージョン名。
platform.azure.zone	マシンを配置するアベイラビリティゾーンの一覧。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。	ゾーンの一覧 (例: ["1", "2", "3"])。
platform.azure.networkResourceGroupName	クラスターをデプロイする既存の VNet を含むリソースグループの名前。この名前は platform.azure.baseDomainResourceGroupName と同じにすることはできません。	文字列。
platform.azure.virtualNetwork	クラスターをデプロイする既存 VNet の名前。	文字列。

パラメーター	説明	値
platform.azure.controlPlaneSubnet	コントロールプレーンマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。
platform.azure.computeSubnet	コンピューターマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。



注記

Azure クラスタで、[Azure アベイラビリティゾーン](#) のカスタマイズや [タグ](#)を使用した [Azure リソースの編成](#) を実行することはできません。

1.3.5.2. Azure のカスタマイズされた `install-config.yaml` ファイルのサンプル

`install-config.yaml` ファイルをカスタマイズして、OpenShift Container Platform クラスタのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。



重要

このサンプルの YAML ファイルは参照用にのみ提供されます。インストールプログラムを使用して `install-config.yaml` ファイルを取得し、これを変更する必要があります。

```

apiVersion: v1
baseDomain: example.com ❶
controlPlane: ❷
  hyperthreading: Enabled ❸ ❹
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 ❺
        type: Standard_D8s_v3
      replicas: 3
compute: ❻
- hyperthreading: Enabled ❼
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 ❽
      zones: ❾
      - "1"
      - "2"
      - "3"
    replicas: 5
metadata:
  name: test-cluster ❿

```

```

networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
pullSecret: '{"auths": ...}' 13
ifndef::openshift-origin
fips: false 14
sshKey: ssh-ed25519 AAAA... 15
endif::openshift-origin
ifdef::openshift-origin
sshKey: ssh-ed25519 AAAA... 16
endif::openshift-origin

```

1 10 11 13 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

2 6 これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

3 7 **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン - で始め、**controlPlane** セクションの最初の行はハイフンで始めることができます。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1つのコントロールプレーンプールのみが使用されます。

4 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。



重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。同時マルチスレッドを無効にする場合は、マシンに対して **Standard_D8s_v3** などの大規模な仮想マシンタイプを使用します。

5 8 使用するディスクのサイズは、GB 単位で指定できます。マスターノードの最小推奨値は 1024 GB です。

9 マシンをデプロイするゾーンの一覧を指定します。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。

12 ベースドメインの DNS ゾーンが含まれるリソースグループの名前を指定します。

14 16 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされま

ん。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。

- 15 クラスター内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.3.6. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に1回だけ実行できます。

前提条件

- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

```
$ ./openshift-install create cluster --dir=<installation_directory> \ 1
--log-level=info 2
```

- 1 **<installation_directory>** については、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。

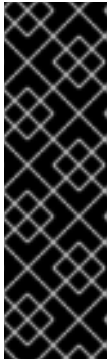
- 2 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



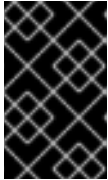
注記

ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。

**重要**

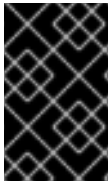
インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

**重要**

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

1.3.7. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。

**重要**

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.3.7.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。**PATH** を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.3.7.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.3.7.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.3.8. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。
- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig ❶
```

- ❶ **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

1.3.9. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

1.4. ネットワークのカスタマイズによる AZURE へのクラスターのインストール

OpenShift Container Platform バージョン 4.5 では、インストールプログラムが Microsoft Azure にプロビジョニングするインフラストラクチャーにカスタマイズされたネットワーク設定でクラスターをインストールできます。ネットワーク設定をカスタマイズすることにより、クラスターは環境内の既存の IP アドレスの割り当てと共存でき、既存の MTU および VXLAN 設定と統合できます。

大半のネットワーク設定パラメーターはインストール時に設定する必要があり、実行中のクラスターで変更できるのは **kubeProxy** 設定パラメーターのみになります。

1.4.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。

- クラスターをホストできるように [Azure アカウントを設定](#) し、クラスターをデプロイするテスト済みの検証されたリージョンを判別します。
- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。

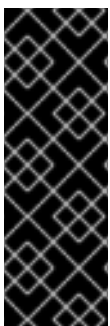
1.4.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.4.3. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が `~/.ssh` ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。



注記

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ①
```

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.4.4. インストールプログラムの取得

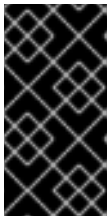
OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

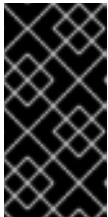
手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.4.5. インストール設定ファイルの作成

Microsoft Azure にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. **install-config.yaml** ファイルを作成します。

- a. 以下のコマンドを実行します。

```
$ ./openshift-install create install-config --dir=<installation_directory> 1
```

- 1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。



重要

空のディレクトリーを指定します。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

- b. プロンプト時に、クラウドの設定の詳細情報を指定します。

- i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

- ii. ターゲットに設定するプラットフォームとして **azure** を選択します。
- iii. お使いのコンピューターに Microsoft Azure プロファイルが保存されていない場合は、サブスクリプションとサービスプリンシパルに以下の Azure パラメーター値を指定します。
 - **azure subscription id** クラスターに使用するサブスクリプション ID。アカウント出力に **id** 値を指定します。
 - **azure tenant id** テナント ID。アカウント出力に **tenantId** 値を指定します。
 - **azure service principal client id** サービスプリンシパルの **appId** パラメーターの値。

- **azure service principal client secret** サービスプリンシパルの **password** パラメーターの値。
- iv. クラスターをデプロイするリージョンを選択します。
- v. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成した Azure DNS ゾーンに対応します。
- vi. クラスターの記述名を入力します。



重要

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

- vii. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから取得したプルシークレットを貼り付けます。
- 2. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細については、**インストール設定パラメーター**セクションを参照してください。
- 3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。



重要

install-config.yaml ファイルはインストールプロセス時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

1.4.5.1. インストール設定パラメーター

OpenShift Container Platform クラスターをデプロイする前に、クラスターをホストするクラウドプラットフォームでアカウントを記述し、クラスターのプラットフォームをオプションでカスタマイズするためにパラメーターの値を指定します。**install-config.yaml** インストール設定ファイルを作成する際に、コマンドラインで必要なパラメーターの値を指定します。クラスターをカスタマイズする場合、**install-config.yaml** ファイルを変更して、プラットフォームについての詳細情報を指定できます。



注記

インストール後は、これらのパラメーターを **install-config.yaml** ファイルで変更することはできません。



重要

openshift-install コマンドは、パラメーターのフィールド名を検証しません。正しくない名前を指定すると、関連するファイルまたはオブジェクトは作成されず、エラーが報告されません。指定されたパラメーターのフィールド名が正しいことを確認します。

1.4.5.1.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表1.5 必須パラメーター

パラメーター	説明	値
apiVersion	install-config.yaml コンテンツの API バージョン。現在のバージョンは v1 です。インストーラーは、古い API バージョンをサポートすることもできます。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、 baseDomain と <metadata.name> 、 <baseDomain> 形式を使用する metadata.name パラメーターの値の組み合わせです。	example.com などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース ObjectMeta 。ここからは name パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて {{.metadata.name}} 、 {{.baseDomain}} のサブドメインです。	dev などの小文字、ハイフン (-)、およびピリオド (.) が含まれる文字列。
platform	インストールの実行に使用する特定プラットフォームの設定: aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。 platform.<platform> パラメーターに関する追加情報は、以下の表で特定のプラットフォームについて参照してください。	オブジェクト

パラメーター	説明	値
pullSecret	https://cloud.redhat.com/openshift/install/pull-secret からプルシークレットを取得し、Quay.io などのサービスから OpenShift Container Platform コンポーネントのコンテナイメージのダウンロードを認証します。	<pre>{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }</pre>

1.4.5.1.2. ネットワーク設定パラメーター

既存のネットワークインフラストラクチャーの要件に基づいて、インストール設定をカスタマイズできます。たとえば、クラスターネットワークの IP アドレスブロックを拡張するか、デフォルトとは異なる IP アドレスブロックを指定できます。

IPv4 アドレスのみがサポートされます。

表1.6 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	オブジェクト  注記 インストール後に networking オブジェクトで指定したパラメーターを変更することはできません。
networking.networkType	インストールするクラスターネットワークプロバイダー Container Network Interface (CNI) プラグイン。	OpenShiftSDN または OVNKubernetes のいずれか。デフォルト値は OpenShiftSDN です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は 10.128.0.0/14 で、ホストの接頭辞は /23 です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>

パラメーター	説明	値
networking.clusterNetwork.cidr	<p>networking.clusterNetwork を使用する場合に必須です。IP アドレスブロック。</p> <p>IPv4 ネットワーク</p>	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は 0 から 32 の間になります。
networking.clusterNetwork.hostPrefix	<p>それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます。hostPrefix 値の 23 は、$2^{(32-23)} - 2$ Pod IP アドレスを提供します。</p>	<p>サブネット接頭辞。</p> <p>デフォルト値は 23 です。</p>
networking.serviceNetwork	<p>サービスの IP アドレスブロック。デフォルト値は 172.30.0.0/16 です。</p> <p>OpenShift SDN および OVN-Kubernetes ネットワークプロバイダーは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。</p>	<p>CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。</p> <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
networking.machineNetwork	<p>マシンの IP アドレスブロック。</p> <p>複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。</p>	<p>オブジェクトの配列。以下に例を示します。</p> <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
networking.machineNetwork.cidr	<p>networking.machineNetwork を使用する場合に必須です。IP アドレスブロック。libvirt 以外のすべてのプラットフォームでは、デフォルト値は 10.0.0.0/16 です。libvirt の場合、デフォルト値は 192.168.126.0/24 です。</p>	<p>CIDR 表記の IP ネットワークブロック。</p> <p>例: 10.0.0.0/16</p> <div>  <div> <p>注記</p> <p>優先される NIC が置かれている CIDR に一致する networking.machineNetwork を設定します。</p> </div> </div>

1.4.5.1.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表1.7 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定される際にも使用できます。	文字列
compute	コンピュートノードを設定するマシンの設定。	machine-pool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
compute.hyperthreading	<p>コンピュートマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div data-bbox="491 1216 600 1503" data-label="Image"> </div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p>	Enabled または Disabled
compute.name	compute を使用する場合に必須です。マシンプールの名前。	worker
compute.platform	compute を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は controlPlane.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 ovirt 、 vsphere 、または {}
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	2 以上の正の整数。デフォルト値は 3 です。

パラメーター	説明	値
controlPlane	コントロールプレーンを設定するマシンの設定。	MachinePool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
controlPlane.hyperthreading	<p>コントロールプレーンマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div>  <div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p> </div> </div>	Enabled または Disabled
controlPlane.name	controlPlane を使用する場合に必須です。マシンプールの名前。	master
controlPlane.platform	controlPlane を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は compute.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は 3 のみです (これはデフォルト値です)。

パラメーター	説明	値
fips	<p>FIPS モードを有効または無効にします。デフォルトは false (無効) です。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。</p> <div>  <div> 注記 <p>Azure File ストレージを使用している場合、FIPS モードを有効にすることはできません。</p> </div> </div>	false または true
imageContentSources	release-image コンテンツのソースおよびリポジトリ。	オブジェクトの配列。この表の以下の行で説明されているように、 source およびオプションで mirrors が含まれます。
imageContentSources.source	imageContentSources を使用する場合に必須です。ユーザーが参照するリポジトリを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリを1つ以上指定します。	文字列の配列。
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	Internal または External 。プライベートクラスターをデプロイするには、 publish を Internal に設定します。これはインターネットからアクセスできません。デフォルト値は External です。

パラメーター	説明	値
sshKey	<p>クラスターマシンへのアクセスを認証するための SSH キー。</p> <div>  <div> <p>注記</p> <p>インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、ssh-agent プロセスが使用する SSH キーを指定します。</p> </div> </div>	たとえば、 sshKey: ssh-ed25519 AAAA.. です。

1.4.5.1.4. 追加の Azure 設定パラメーター

追加の Azure 設定パラメーターは以下の表で説明されています。

表1.8 追加の Azure パラメーター

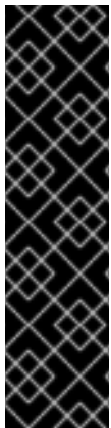
パラメーター	説明	値
controlPlane.platform.azure.osDisk.diskSizeGB	VM の Azure ディスクのサイズ。	GB 単位でディスクのサイズを表す整数。サポートされる最小のディスクサイズは 1024 です。
platform.azure.baseDomainResourceGroupName	ベースドメインの DNS ゾーンが含まれるリソースグループの名前。	文字列 (例: production_cluster)。
platform.azure.region	クラスターをホストする Azure リージョンの名前。	centralus などの有効なリージョン名。
platform.azure.zone	マシンを配置するアベイラビリティゾーンの一覧。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。	ゾーンの一覧 (例: ["1", "2", "3"])。
platform.azure.networkResourceGroupName	クラスターをデプロイする既存の VNet を含むリソースグループの名前。この名前は platform.azure.baseDomainResourceGroupName と同じにすることはできません。	文字列。
platform.azure.virtualNetwork	クラスターをデプロイする既存 VNet の名前。	文字列。

パラメーター	説明	値
platform.azure.controlPlaneSubnet	コントロールプレーンマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。
platform.azure.computeSubnet	コンピュートマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。



注記

Azure クラスターで、[Azure アベイラビリティゾーン](#) のカスタマイズや [タグ](#)を使用した [Azure リソースの編成](#) を実行することはできません。



重要

Open Virtual Networking (OVN) Kubernetes ネットワークプラグインは、テクノロジープレビュー機能です。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

OVN テクノロジープレビュー機能のサポート範囲についての詳細は、<https://access.redhat.com/articles/4380121> を参照してください。

1.4.5.2. ネットワーク設定パラメーター

クラスターのネットワーク設定パラメーターは **install-config.yaml** 設定ファイルで変更できます。以下の表では、これらのパラメーターについて説明しています。



注記

インストール後は、**install-config.yaml** ファイルでこれらのパラメーターを変更することはできません。

表1.9 必要なネットワークパラメーター

パラメーター	説明	値
networking.networkType	デプロイするデフォルトの Container Network Interface (CNI) ネットワークプロバイダープラグイン。 OpenShiftSDN プラグインのみが OpenShift Container Platform 4.5 でサポートされているプラグインです。 OVNKubernetes プラグインは、OpenShift Container Platform 4.5 でテクノロジープレビューとしてご利用いただけます。	OpenShiftSDN または OVNKubernetes のいずれか。デフォルト値は OpenShiftSDN です。

パラメーター	説明	値
networking.clusterNetwork[].cidr	Pod IP アドレスの割り当てに使用する IP アドレスのブロック。 OpenShiftSDN ネットワークプラグインは複数のクラスターネットワークをサポートします。複数のクラスターネットワークのアドレスブロックには重複が許可されません。予想されるワークロードに適したサイズのアドレスプールを選択してください。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 10.128.0.0/14 です。
networking.clusterNetwork[].hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞の長さ。たとえば、 hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます (510 ($2^{(32-23)} - 2$) Pod IP アドレスが許可されます)。	サブネット接頭辞。デフォルト値は 23 です。
networking.serviceNetwork[]	サービスの IP アドレスのブロック。 OpenShiftSDN は1つの serviceNetwork ブロックのみを許可します。このアドレスブロックは他のネットワークブロックと重複できません。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 172.30.0.0/16 です。
networking.machineNetwork[].cidr	クラスターのインストール中に OpenShift Container Platform インストールプログラムによって使用されるノードに割り当てられる IP アドレスのブロック。このアドレスブロックは他のネットワークブロックと重複できません。複数の CIDR 範囲を指定できます。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 10.0.0.0/16 です。

1.4.5.3. Azure のカスタマイズされた install-config.yaml ファイルのサンプル

install-config.yaml ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。



重要

このサンプルの YAML ファイルは参照用에만提供されます。インストールプログラムを使用して **install-config.yaml** ファイルを取得し、これを変更する必要があります。

```

apiVersion: v1
baseDomain: example.com ❶
controlPlane: ❷
hyperthreading: Enabled ❸ ❹
name: master
platform:
  azure:
    osDisk:
      diskSizeGB: 1024 ❺
      type: Standard_D8s_v3
  replicas: 3
compute: ❻

```

```

- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
        - "1"
        - "2"
        - "3"
    replicas: 5
  metadata:
    name: test-cluster 10
  networking: 11
    clusterNetwork:
      - cidr: 10.128.0.0/14
        hostPrefix: 23
    machineNetwork:
      - cidr: 10.0.0.0/16
    networkType: OpenShiftSDN
    serviceNetwork:
      - 172.30.0.0/16
  platform:
    azure:
      region: centralus 12
      baseDomainResourceGroupName: resource_group 13
  pullSecret: '{"auths": ...}' 14
  ifndef::openshift-origin
  fips: false 15
  sshKey: ssh-ed25519 AAAA... 16
  endif::openshift-origin
  ifdef::openshift-origin
  sshKey: ssh-ed25519 AAAA... 17
  endif::openshift-origin

```

1 10 12 14 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

2 6 11 これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

3 7 **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン - で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1つのコントロールプレーンプールのみが使用されます。

4 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。



重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。同時マルチスレッドを無効にする場合は、マシンに対して **Standard_D8s_v3** などの大規模な仮想マシンタイプを使用します。

- 5 8 使用するディスクのサイズは、GB 単位で指定できます。マスターノードの最小推奨値は 1024 GB です。
- 9 マシンをデプロイするゾーンの一覧を指定します。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。
- 13 ベースドメインの DNS ゾーンが含まれるリソースグループの名前を指定します。
- 15 17 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。
- 16 クラスタ内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスタでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.4.6. 高度なネットワーク設定パラメーターの変更

高度なネットワーク設定パラメーターは、クラスタのインストール前にのみ変更することができます。高度な設定のカスタマイズにより、クラスタを既存のネットワーク環境に統合させることができます。これを実行するには、MTU または VXLAN ポートを指定し、[kube-proxy](#) 設定のカスタマイズを許可し、**openshiftSDNConfig** パラメーターに異なる **mode** を指定します。



重要

インストールプログラムで作成される OpenShift Container Platform マニフェストファイルの変更はサポートされていません。以下の手順のように、作成するマニフェストファイルを適用することがサポートされています。

前提条件

- **install-config.yaml** ファイルを作成し、これに対する変更を完了します。

手順

1. 以下のコマンドを使用してマニフェストを作成します。

```
$ ./openshift-install create manifests --dir=<installation_directory> 1
```

- 1 **<installation_directory>** については、クラスタの **install-config.yaml** ファイルが含まれるディレクトリーの名前を指定します。

2. **cluster-network-03-config.yml** という名前のファイルを **<installation_directory>/manifests/** ディレクトリーに作成します。

```
$ touch <installation_directory>/manifests/cluster-network-03-config.yml ❶
```

- ❶ **<installation_directory>** については、クラスターの **manifests/** ディレクトリーが含まれるディレクトリー名を指定します。

ファイルの作成後は、以下のようにいくつかのネットワーク設定ファイルが **manifests/** ディレクトリーに置かれます。

```
$ ls <installation_directory>/manifests/cluster-network-*
```

出力例

```
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

3. エディターで **cluster-network-03-config.yml** ファイルを開き、必要な Operator 設定を記述する CR を入力します。

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec: ❶
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  serviceNetwork:
    - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
```

- ❶ **spec** パラメーターのパラメーターは例です。CR に Cluster Network Operator の設定を指定します。

CNO は CR にパラメーターのデフォルト値を提供するため、変更が必要なパラメーターのみを指定する必要があります。

4. **cluster-network-03-config.yml** ファイルを保存し、テキストエディターを終了します。
5. オプション: **manifests/cluster-network-03-config.yml** ファイルをバックアップします。インストールプログラムは、クラスターの作成時に **manifests/** ディレクトリーを削除します。

1.4.7. Cluster Network Operator (CNO) の設定

クラスターネットワークの設定は、Cluster Network Operator (CNO) 設定の一部として指定され、**cluster** という名前の CR オブジェクトに保存されます。CR は **operator.openshift.io** API グループの **Network** API のパラメーターを指定します。

defaultNetwork パラメーターのパラメーター値を CNO CR に設定することにより、OpenShift Container Platform クラスターのクラスターネットワーク設定を指定できます。以下の CR は、CNO のデフォルト設定を表示し、設定可能なパラメーターと有効なパラメーターの値の両方について説明しています。

Cluster Network Operator CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork: ❶
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork: ❷
  - 172.30.0.0/16
  defaultNetwork: ❸
  ...
  kubeProxyConfig: ❹
    iptablesSyncPeriod: 30s ❺
    proxyArguments:
      iptables-min-sync-period: ❻
      - 0s
```

❶❷ **install-config.yaml** ファイルに指定されます。

❸ クラスターネットワークのデフォルトの Container Network Interface (CNI) ネットワークプロバイダーを設定します。

❹ このオブジェクトのパラメーターは、**kube-proxy** 設定を指定します。パラメーターの値を指定しない場合、クラスターネットワーク Operator は表示されるデフォルトのパラメーター値を適用します。OVN-Kubernetes デフォルト CNI ネットワークプロバイダーを使用している場合、**kube-proxy** 設定は機能しません。

❺ **iptables** ルールの更新期間。デフォルト値は **30s** です。有効な接尾辞には、**s**、**m**、および **h** などが含まれ、これらについては、[Go Package time](#) ドキュメントで説明されています。



注記

OpenShift Container Platform 4.3 以降で強化されたパフォーマンスの向上により、**iptablesSyncPeriod** パラメーターを調整する必要はなくなりました。

❻ **iptables** ルールを更新する前の最小期間。このパラメーターにより、更新の頻度が高くなり過ぎないようにできます。有効な接尾辞には、**s**、**m**、および **h** などが含まれ、これらについては、[Go Package time](#) で説明されています。

1.4.7.1. OpenShift SDN デフォルト CNI ネットワークプロバイダーの設定パラメーター

以下の YAML オブジェクトは、OpenShift SDN デフォルト Container Network Interface (CNI) ネットワークプロバイダーの設定パラメーターについて説明しています。

```
defaultNetwork:
  type: OpenShiftSDN ❶
  openshiftSDNConfig: ❷
    mode: NetworkPolicy ❸
    mtu: 1450 ❹
    vxlanPort: 4789 ❺
```

- ❶ **install-config.yaml** ファイルに指定されます。
- ❷ OpenShift SDN 設定の一部を上書きする必要がある場合にのみ指定します。
- ❸ OpenShift SDN のネットワーク分離モードを設定します。許可される値は **Multitenant**、**Subnet**、または **NetworkPolicy** です。デフォルト値は **NetworkPolicy** です。
- ❹ VXLAN オーバーレイネットワークの最大転送単位 (MTU)。これは、プライマリーネットワークインターフェイスの MTU に基づいて自動的に検出されます。通常、検出された MTU を上書きする必要はありません。

自動検出した値が予想される値ではない場合は、ノード上のプライマリーネットワークインターフェイスの MTU が正しいことを確認します。このオプションを使用して、ノード上のプライマリーネットワークインターフェイスの MTU 値を変更することはできません。

クラスターで異なるノードに異なる MTU 値が必要な場合、この値をクラスター内の最小の MTU 値よりも **50** 小さく設定する必要があります。たとえば、クラスター内の一部のノードでは MTU が **9001** であり、MTU が **1500** のクラスターもある場合には、この値を **1450** に設定する必要があります。

- ❺ すべての VXLAN パケットに使用するポート。デフォルト値は **4789** です。別の VXLAN ネットワークの一部である既存ノードと共に仮想化環境で実行している場合は、これを変更する必要がある可能性があります。たとえば、OpenShift SDN オーバーレイを VMware NSX-T 上で実行する場合は、両方の SDN が同じデフォルトの VXLAN ポート番号を使用するため、VXLAN の別のポートを選択する必要があります。

Amazon Web Services (AWS) では、VXLAN にポート **9000** とポート **9999** 間の代替ポートを選択できます。

1.4.7.2. OVN-Kubernetes デフォルト CNI ネットワークプロバイダーの設定パラメーター

以下の YAML オブジェクトは OVN-Kubernetes デフォルト CNI ネットワークプロバイダーの設定パラメーターについて説明しています。

```
defaultNetwork:
  type: OVNKubernetes ❶
  ovnKubernetesConfig: ❷
    mtu: 1400 ❸
    genevePort: 6081 ❹
```

- ❶ **install-config.yaml** ファイルに指定されます。
- ❷ OVN-Kubernetes 設定の一部を上書きする必要がある場合にのみ指定します。

- 3 Geneve (Generic Network Virtualization Encapsulation) オーバーレイネットワークの MTU (maximum transmission unit)。これは、プライマリーネットワークインターフェイスの MTU に基

自動検出した値が予想される値ではない場合は、ノード上のプライマリーネットワークインターフェイスの MTU が正しいことを確認します。このオプションを使用して、ノード上のプライマリーネットワークインターフェイスの MTU 値を変更することはできません。

クラスターで異なるノードに異なる MTU 値が必要な場合、この値をクラスター内の最小の MTU 値よりも **100** 小さく設定する必要があります。たとえば、クラスター内の一部のノードでは MTU が **9001** であり、MTU が **1500** のクラスターもある場合には、この値を **1400** に設定する必要があります。

- 4 Geneve オーバーレイネットワークの UDP ポート。

1.4.7.3. Cluster Network Operator の設定例

以下の例のように、CNO の完全な CR オブジェクトが表示されます。

Cluster Network Operator のサンプル CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  serviceNetwork:
    - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
  kubeProxyConfig:
    iptablesSyncPeriod: 30s
    proxyArguments:
      iptables-min-sync-period:
        - 0s
```

1.4.8. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に 1 回だけ実行できます。

前提条件

- クラスターをホストするクラウドプラットフォームでアカウントを設定します。

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

```
$ ./openshift-install create cluster --dir=<installation_directory> \ ❶  
--log-level=info ❷
```

❶ **<installation_directory>** については、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。

❷ 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



注記

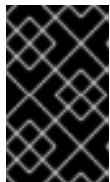
ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、**kubelet** 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。



重要

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

1.4.9. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。



重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.4.9.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvfz <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.4.9.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.4.9.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATHを確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.4.10. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。
- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig ❶
```

- ❶ **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

1.4.11. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

1.5. AZURE のクラスターの既存 VNET へのインストール

OpenShift Container Platform バージョン 4.5 では、クラスターを Microsoft Azure の既存の Azure Virtual Network (VNet) にインストールできます。インストールプログラムは、カスタマイズ可能な残りの必要なインフラストラクチャーをプロビジョニングします。インストールをカスタマイズするには、クラスターをインストールする前に、**install-config.yaml** ファイルでパラメーターを変更します。

1.5.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- クラスターをホストできるように [Azure アカウントを設定](#) し、クラスターをデプロイするテスト済みの検証されたリージョンを判別します。
- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。

1.5.2. OpenShift Container Platform クラスターでの VNet の再利用について

OpenShift Container Platform 4.5 では、クラスターを Microsoft Azure の既存の Azure Virtual Network (VNet) にデプロイできます。これを実行する場合、VNet 内の既存のサブネットおよびルーティングルールも使用する必要があります。

OpenShift Container Platform を既存の Azure VNet にデプロイすることで、新規アカウントでのサービス制限の制約を回避したり、会社のガイドラインによる運用上の制約をより容易に遵守することが可能になる場合があります。VNet の作成に必要なインフラストラクチャーの作成パーミッションを取得できない場合には、このオプションを使用できます。



重要

既存の VNet を使用するには、更新された Azure Private DNS (プレビュー) 機能を使用する必要があります。この機能の制限についての詳細は、[Announcing Preview Refresh for Azure DNS Private Zones](#) を参照してください。

1.5.2.1. VNet を使用するための要件

既存の VNet を使用してクラスターをデプロイする場合、クラスターをインストールする前に追加のネットワーク設定を実行する必要があります。インストーラーでプロビジョニングされるインフラストラクチャークラスターでは、インストーラーは通常以下のコンポーネントを作成しますが、既存の VNet にインストールする場合にはこれらを作成しません。

- サブネット
- ルートテーブル
- VNets
- ネットワークセキュリティグループ

カスタム VNet を使用する場合、インストールプログラムおよびクラスターでできるようにカスタム VNet およびそのサブネットを適切に設定する必要があります。インストールプログラムは、使用するクラスターのネットワーク範囲を細分化できず、サブネットのルートテーブルを設定するか、または DHCP などの VNet オプションを設定します。これは、クラスターのインストール前に設定する必要があります。

クラスターは、既存の VNet およびサブネットを含むリソースグループにアクセスする必要があります。クラスターが作成するすべてのリソースは、作成される別個のリソースグループに配置され、一部のネットワークリソースが別個のグループから使用されます。一部のクラスター Operator は両方のリソースグループのリソースにアクセスする必要があります。たとえば マシン API コントローラーは、ネットワークリソースグループから、作成される仮想マシンの NIC をサブネットに割り当てます。

VNet には以下の特徴が確認される必要があります。

- VNet の CIDR ブロックには、クラスターマシンの IP アドレスプールである **Networking.MachineCIDR** 範囲が含まれる必要があります。
- VNet およびそのサブネットは同じリソースグループに属する必要があり、サブネットは静的 IP アドレスではなく、Azure で割り当てられた DHCP IP アドレスを使用するように設定される必要があります。

コントロールプレーンマシンのサブネットおよびコンピューティングマシンのサブネットの 2 つのサブネットを VNet 内に指定する必要があります。Azure はマシンを指定するリージョン内の複数の異なるアベイラビリティゾーンに分散するため、デフォルトのクラスターには高可用性があります。

指定するサブネットが適切であることを確認するには、インストールプログラムが以下のデータを確認します。

- 指定したサブネットすべてが存在します。
- コントロールプレーンマシンのサブネットおよびコンピューティングマシンのサブネットの 2 つのサブネットを指定する必要があります。
- サブネットの CIDR は指定されたマシン CIDR に属します。マシンは、プライベートサブネットを指定しないアベイラビリティゾーンにはプロビジョニングされません。必要な場合に、インストールプログラムはコントロールプレーンおよびワーカーノードを管理するパブリックロードバランサーを作成し、Azure はパブリック IP アドレスをそれらに割り当てます。

既存の VNet を使用するクラスターを破棄しても、VNet は削除されません。

1.5.2.1.1. ネットワークセキュリティグループの要件

コンピュータマシンおよびコントロールプレーンマシンをホストするサブネットのネットワークセキュリティグループには、クラスターの通信が正しいことを確認するための特定のアクセスが必要です。必要なクラスター通信ポートへのアクセスを許可するルールを作成する必要があります。



重要

ネットワークセキュリティグループルールは、クラスターのインストール前に有効にされている必要があります。必要なアクセスなしにクラスターのインストールを試行しても、インストールプログラムは Azure API に到達できず、インストールに失敗します。

表1.10 必須ポート

ポート	説明	コントロールプレーン	コンピュータ
80	HTTP トラフィックを許可します。		x
443	HTTPS トラフィックを許可します		x
6443	コントロールプレーンマシンとの通信を許可します。	x	
22623	マシン設定サーバーとの通信を許可します。	x	



注記

クラスターコンポーネントは、Kubernetes コントローラーが更新する、ユーザーによって提供されるネットワークセキュリティグループを変更しないため、擬似セキュリティグループが環境の残りの部分に影響を及ぼさずに Kubernetes コントローラー用に作成されます。

1.5.2.2. パーMISSIONの区分

OpenShift Container Platform 4.3 以降、クラスターのデプロイに、インストールプログラムがプロビジョニングするインフラストラクチャクラスターに必要なすべてのパーミッションを必要としなくなりました。この変更は、ある会社で個人がクラウドで他とは異なるリソースを作成できるようにパーミッションが区分された状態に類似するものです。たとえば、インスタンス、ストレージ、ロードバランサーなどのアプリケーション固有のアイテムを作成することはできますが、VNet、サブネット、または Ingress ルールなどのネットワーク関連のコンポーネントは作成できない可能性があります。

クラスターの作成時に使用する Azure の認証情報には、VNet、およびサブネット、ルーティングテーブル、インターネットゲートウェイ、NAT、VPN などの VNet 内のコアとなるネットワークコンポーネントの作成に必要なネットワークのパーミッションは必要ありません。ロードバランサー、セキュリティグループ、ストレージアカウントおよびノードなどの、クラスター内でマシンに必要なアプリケーションリソースを作成するパーミッションは依然として必要になります。

1.5.2.3. クラスター間の分離

クラスターは既存のサブネットのネットワークセキュリティグループを変更できないため、VNet でクラスターを相互に分離する方法はありません。

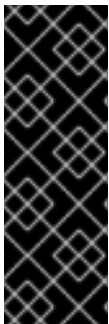
1.5.3. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.5.4. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> ❶
```

- ❶ ~/.ssh/id_rsa などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が ~/.ssh ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。



注記

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ❶
```

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ❶ ~/.ssh/id_rsa などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.5.5. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

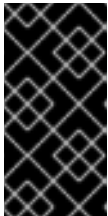
手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.5.6. インストール設定ファイルの作成

Microsoft Azure にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. `install-config.yaml` ファイルを作成します。

- a. 以下のコマンドを実行します。

```
$ ./openshift-install create install-config --dir=<installation_directory> 1
```

- 1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。



重要

空のディレクトリーを指定します。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

- b. プロンプト時に、クラウドの設定の詳細情報を指定します。

- i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。



注記

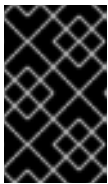
インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

- ii. ターゲットに設定するプラットフォームとして **azure** を選択します。
- iii. お使いのコンピューターに Microsoft Azure プロファイルが保存されていない場合は、サブスクリプションとサービスプリンシパルに以下の Azure パラメーター値を指定します。
 - **azure subscription id** クラスターに使用するサブスクリプション ID。アカウント出力に **id** 値を指定します。
 - **azure tenant id** テナント ID。アカウント出力に **tenantId** 値を指定します。
 - **azure service principal client id** サービスプリンシパルの **appId** パラメーターの値。
 - **azure service principal client secret** サービスプリンシパルの **password** パラメーターの値。
- iv. クラスターをデプロイするリージョンを選択します。
- v. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成した Azure DNS ゾーンに対応します。
- vi. クラスターの記述名を入力します。

**重要**

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

- vii. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから取得したプルシークレットを貼り付けます。
2. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細については、[インストール設定パラメーターセクション](#)を参照してください。
3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。

**重要**

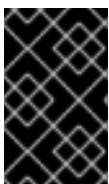
install-config.yaml ファイルはインストールプロセス時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

1.5.6.1. インストール設定パラメーター

OpenShift Container Platform クラスターをデプロイする前に、クラスターをホストするクラウドプラットフォームでアカウントを記述し、クラスターのプラットフォームをオプションでカスタマイズするためにパラメーターの値を指定します。**install-config.yaml** インストール設定ファイルを作成する際に、コマンドラインで必要なパラメーターの値を指定します。クラスターをカスタマイズする場合、**install-config.yaml** ファイルを変更して、プラットフォームについての詳細情報を指定できます。

**注記**

インストール後は、これらのパラメーターを **install-config.yaml** ファイルで変更することはできません。

**重要**

openshift-install コマンドは、パラメーターのフィールド名を検証しません。正しくない名前を指定すると、関連するファイルまたはオブジェクトは作成されず、エラーが報告されません。指定されたパラメーターのフィールド名が正しいことを確認します。

1.5.6.1.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表1.11 必須パラメーター

パラメーター	説明	値
--------	----	---

パラメーター	説明	値
apiVersion	install-config.yaml コンテンツの API バージョン。現在のバージョンは v1 です。インストーラーは、古い API バージョンをサポートすることもできます。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、 baseDomain と <metadata.name> 。 <baseDomain> 形式を使用する metadata.name パラメーターの値の組み合わせです。	example.com などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース ObjectMeta 。ここからは name パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて {{.metadata.name}} 。 {{.baseDomain}} のサブドメインです。	dev などの小文字、ハイフン (-)、およびピリオド (.) が含まれる文字列。
platform	インストールの実行に使用する特定プラットフォームの設定: aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。 platform.<platform> パラメーターに関する追加情報は、以下の表で特定のプラットフォームについて参照してください。	オブジェクト

パラメーター	説明	値
pullSecret	https://cloud.redhat.com/openshift/install/pull-secret からプルシークレットを取得し、Quay.io などのサービスから OpenShift Container Platform コンポーネントのコンテナイメージのダウンロードを認証します。	<pre>{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }</pre>

1.5.6.1.2. ネットワーク設定パラメーター

既存のネットワークインフラストラクチャーの要件に基づいて、インストール設定をカスタマイズできます。たとえば、クラスターネットワークの IP アドレスブロックを拡張するか、デフォルトとは異なる IP アドレスブロックを指定できます。

IPv4 アドレスのみがサポートされます。

表1.12 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	<p>オブジェクト</p>  <p>注記</p> <p>インストール後に networking オブジェクトで指定したパラメーターを変更することはできません。</p>
networking.networkType	インストールするクラスターネットワークプロバイダー Container Network Interface (CNI) プラグイン。	OpenShiftSDN または OVNKubernetes のいずれか。デフォルト値は OpenShiftSDN です。
networking.clusterNetwork	<p>Pod の IP アドレスブロック。</p> <p>デフォルト値は 10.128.0.0/14 で、ホストの接頭辞は /23 です。</p> <p>複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。</p>	<p>オブジェクトの配列。以下に例を示します。</p> <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>

パラメーター	説明	値
networking.clusterNetwork.cidr	<p>networking.clusterNetwork を使用する場合に必須です。IP アドレスブロック。</p> <p>IPv4 ネットワーク</p>	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は 0 から 32 の間になります。
networking.clusterNetwork.hostPrefix	<p>それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます。hostPrefix 値の 23 は、$2^{(32 - 23) - 2}$ Pod IP アドレスを提供します。</p>	<p>サブネット接頭辞。</p> <p>デフォルト値は 23 です。</p>
networking.serviceNetwork	<p>サービスの IP アドレスブロック。デフォルト値は 172.30.0.0/16 です。</p> <p>OpenShift SDN および OVN-Kubernetes ネットワークプロバイダーは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。</p>	<p>CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。</p> <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
networking.machineNetwork	<p>マシンの IP アドレスブロック。</p> <p>複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。</p>	<p>オブジェクトの配列。以下に例を示します。</p> <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
networking.machineNetwork.cidr	<p>networking.machineNetwork を使用する場合に必須です。IP アドレスブロック。libvirt 以外のすべてのプラットフォームでは、デフォルト値は 10.0.0.0/16 です。libvirt の場合、デフォルト値は 192.168.126.0/24 です。</p>	<p>CIDR 表記の IP ネットワークブロック。</p> <p>例: 10.0.0.0/16</p> <div>  <div> <p>注記</p> <p>優先される NIC が置かれている CIDR に一致する networking.machineNetwork を設定します。</p> </div> </div>


1.5.6.1.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表1.13 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定される際にも使用できます。	文字列
compute	コンピュートノードを設定するマシンの設定。	machine-pool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
compute.hyperthreading	<p>コンピュートマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div data-bbox="491 1216 600 1500" data-label="Image"> </div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p>	Enabled または Disabled
compute.name	compute を使用する場合に必須です。マシンプールの名前。	worker
compute.platform	compute を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は controlPlane.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	2 以上の正の整数。デフォルト値は 3 です。

パラメーター	説明	値
controlPlane	コントロールプレーンを設定するマシンの設定。	MachinePool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列
controlPlane.hyperthreading	<p>コントロールプレーンマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div>  <div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p> </div> </div>	Enabled または Disabled
controlPlane.name	controlPlane を使用する場合に必須です。マシンプールの名前。	master
controlPlane.platform	controlPlane を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は compute.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は 3 のみです (これはデフォルト値です)。

パラメーター	説明	値
fips	<p>FIPS モードを有効または無効にします。デフォルトは false (無効) です。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。</p> <div>  <div> 注記 <p>Azure File ストレージを使用している場合、FIPS モードを有効にすることはできません。</p> </div> </div>	false または true
imageContentSources	release-image コンテンツのソースおよびリポジトリ。	オブジェクトの配列。この表の以下の行で説明されているように、 source およびオプションで mirrors が含まれます。
imageContentSources.source	imageContentSources を使用する場合に必須です。ユーザーが参照するリポジトリを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリを1つ以上指定します。	文字列の配列。
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	Internal または External 。プライベートクラスターをデプロイするには、 publish を Internal に設定します。これはインターネットからアクセスできません。デフォルト値は External です。

パラメーター	説明	値
sshKey	<p>クラスターマシンへのアクセスを認証するための SSH キー。</p> <div>  <div> <p>注記</p> <p>インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、ssh-agent プロセスが使用する SSH キーを指定します。</p> </div> </div>	たとえば、 sshKey: ssh-ed25519 AAAA.. です。

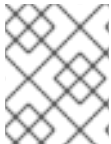
1.5.6.1.4. 追加の Azure 設定パラメーター

追加の Azure 設定パラメーターは以下の表で説明されています。

表1.14 追加の Azure パラメーター

パラメーター	説明	値
controlPlane.platform.azure.osDisk.diskSizeGB	VM の Azure ディスクのサイズ。	GB 単位でディスクのサイズを表す整数。サポートされる最小のディスクサイズは 1024 です。
platform.azure.baseDomainResourceGroupName	ベースドメインの DNS ゾーンが含まれるリソースグループの名前。	文字列 (例: production_cluster)。
platform.azure.region	クラスターをホストする Azure リージョンの名前。	centralus などの有効なリージョン名。
platform.azure.zone	マシンを配置するアベイラビリティゾーンの一覧。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。	ゾーンの一覧 (例: ["1", "2", "3"])。
platform.azure.networkResourceGroupName	クラスターをデプロイする既存の VNet を含むリソースグループの名前。この名前は platform.azure.baseDomainResourceGroupName と同じにすることはできません。	文字列。
platform.azure.virtualNetwork	クラスターをデプロイする既存 VNet の名前。	文字列。

パラメーター	説明	値
platform.azure.controlPlaneSubnet	コントロールプレーンマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。
platform.azure.computeSubnet	コンピュータマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。

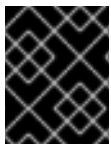


注記

Azure クラスターで、[Azure アベイラビリティゾーン](#)のカスタマイズや [タグ](#)を使用した [Azure リソースの編成](#) を実行することはできません。

1.5.6.2. Azure のカスタマイズされた `install-config.yaml` ファイルのサンプル

`install-config.yaml` ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。



重要

このサンプルの YAML ファイルは参照用にのみ提供されます。インストールプログラムを使用して `install-config.yaml` ファイルを取得し、これを変更する必要があります。

```

apiVersion: v1
baseDomain: example.com ❶
controlPlane: ❷
  hyperthreading: Enabled ❸ ❹
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 ❺
        type: Standard_D8s_v3
      replicas: 3
compute: ❻
- hyperthreading: Enabled ❼
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 ❽
      zones: ❾
      - "1"
      - "2"
      - "3"
    replicas: 5
metadata:
  name: test-cluster ❿

```

```

networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
    networkResourceGroupName: vnet_resource_group 13
    virtualNetwork: vnet 14
    controlPlaneSubnet: control_plane_subnet 15
    computeSubnet: compute_subnet 16
  pullSecret: '{"auths": ...}' 17
  fips: false 18
  sshKey: ssh-ed25519 AAAA... 19

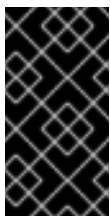
```

1 10 11 17 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

2 6 これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

3 7 **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン - で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1つのコントロールプレーンプールのみが使用されます。

4 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。



重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。同時マルチスレッドを無効にする場合は、マシンに対して **Standard_D8s_v3** などの大規模な仮想マシンタイプを使用します。

5 8 使用するディスクのサイズは、GB 単位で指定できます。マスターノードの最小推奨値は 1024 GB です。

9 マシンをデプロイするゾーンの一覧を指定します。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。

12 ベースドメインの DNS ゾーンが含まれるリソースグループの名前を指定します。

13 既存の VNet を使用する場合は、それが含まれるリソースグループの名前を指定します。

- 14 既存の VNet を使用する場合は、その名前を指定します。
- 15 既存の VNet を使用する場合は、コントロールプレーンマシンをホストするサブネットの名前を指定します。
- 16 既存の VNet を使用する場合は、コンピューティングマシンをホストするサブネットの名前を指定します。
- 18 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。
- 19 クラスター内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.5.6.3. インストール時のクラスター全体のプロキシの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシを使用することができます。プロキシ設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスターをプロキシを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイルが必要です。
- クラスターがアクセスする必要があるサイトを確認し、プロキシをバイパスする必要があるかどうかを判別します。デフォルトで、すべてのクラスター egress トラフィック (クラスターをホストするクラウドについてのクラウドプロバイダー API に対する呼び出しを含む) はプロキシされます。**Proxy** オブジェクトの **spec.noProxy** フィールドにサイトを追加し、必要に応じてプロキシをバイパスします。



注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud Platform (GCP)、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP) へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント (**169.254.169.254**) も設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシ設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
```

```

proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: http://<username>:<pswd>@<ip>:<port> ❷
  noProxy: example.com ❸
  additionalTrustBundle: | ❹
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  ...

```

- ❶ クラスター外の HTTP 接続を作成するために使用するプロキシー URL。URL スキームは **http** である必要があります。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、**httpProxy** 値を指定することはできません。
- ❷ クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。このフィールドが指定されていない場合、HTTP および HTTPS 接続の両方に **httpProxy** が使用されます。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。
- ❸ プロキシーを除外するための宛先ドメイン名、ドメイン、IP アドレス、または他のネットワーク CIDR のコンマ区切りの一覧。サブドメインのみと一致するように、ドメインの前に **.** を付けます。たとえば、**.y.com** は **x.y.com** に一致しますが、**y.com** には一致しません。***** を使用し、すべての宛先のプロキシーをバイパスします。
- ❹ 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティ証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。



注記

インストールプログラムは、プロキシーの **readinessEndpoints** フィールドをサポートしません。

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster Proxy** オブジェクトが依然として作成されますが、これには **spec** がありません。



注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

1.5.7. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に 1 回だけ実行できます。

前提条件

- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

```
$ ./openshift-install create cluster --dir=<installation_directory> \ ❶  
--log-level=info ❷
```

- ❶ **<installation_directory>** については、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。
- ❷ 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



注記

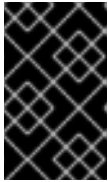
ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、**kubelet** 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

**重要**

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

1.5.8. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。

**重要**

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.5.8.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.5.8.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。

2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.5.8.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.5.9. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。

- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 **<installation_directory>** には、インストールファイルを保存したディレクトリへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

1.5.10. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

1.6. プライベートクラスターの AZURE へのインストール

OpenShift Container Platform バージョン 4.5 では、プライベートクラスターを Microsoft Azure の既存の Azure Virtual Network (VNet) にインストールできます。インストールプログラムは、カスタマイズ可能な残りの必要なインフラストラクチャーをプロビジョニングします。インストールをカスタマイズするには、クラスターをインストールする前に、**install-config.yaml** ファイルでパラメーターを変更します。

1.6.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- クラスターをホストできるように [Azure アカウントを設定](#) し、クラスターをデプロイするテスト済みの検証されたリージョンを判別します。
- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。

1.6.2. プライベートクラスター

お使いの環境で外部のインターネット接続を必要としない場合には、外部エンドポイントを公開しないプライベート OpenShift Container Platform クラスターをデプロイすることができます。プライベートクラスターは内部ネットワークからのみアクセス可能で、インターネット上では表示されません。

デフォルトで、OpenShift Container Platform はパブリックにアクセス可能な DNS およびエンドポイントを使用できるようにプロビジョニングされます。プライベートクラスターは、クラスターのデプロイ時に DNS、Ingress コントローラー、および API サーバーを private に設定します。つまり、クラスターリソースは内部ネットワークからのみアクセスでき、インターネット上では表示されません。

プライベートクラスターをデプロイするには、要件を満たす既存のネットワークを使用する必要があります。クラスターリソースはネットワーク上の他のクラスター間で共有される可能性があります。

さらに、プロビジョニングするクラウドの API サービスにアクセスできるマシンから、プロビジョニングするネットワーク上のホストおよびインストールメディアを取得するために使用するインターネットにプライベートクラスターをデプロイする必要があります。これらのアクセス要件を満たし、所属する会社のガイドラインに準拠したすべてのマシンを使用することができます。たとえば、このマシンには、クラウドネットワーク上の bastion ホスト、または VPN 経由でネットワークにアクセスできるマシンを使用できます。

1.6.2.1. Azure のプライベートクラスター

Microsoft Azure でプライベートクラスターを作成するには、クラスターをホストするために既存のプライベート VNet とサブネットを指定する必要があります。インストールプログラムは、クラスターが必要とする DNS レコードを解決できる必要もあります。インストールプログラムは、内部トラフィック用としてのみ Ingress Operator および API サーバーを設定します。

ネットワークがプライベート VNET に接続される方法によって、クラスターのプライベート DNS レコードを解決するために DNS フォワーダーを使用する必要がある場合があります。クラスターのマシンは、DNS 解決に **168.63.129.16** を内部で使用します。詳細は、Azure ドキュメントの [What is Azure Private DNS?](#) および [What is IP address 168.63.129.16?](#) を参照してください。

クラスターには、Azure API にアクセスするためにインターネットへのアクセスが依然として必要です。

以下のアイテムは、プライベートクラスターのインストール時に必要ではなく、作成されません。

- **BaseDomainResourceGroup** (クラスターがパブリックレコードを作成しないため)
- パブリック IP アドレス
- パブリック DNS レコード
- パブリックエンドポイント

The cluster is configured so that the Operators do not create public records for the cluster and all cluster machines are placed in the private subnets that you specify.

1.6.2.1.1. 制限事項

Azure 上のプライベートクラスターは、既存の VNet の使用に関連する制限のみの制限を受けます。

1.6.3. OpenShift Container Platform クラスターでの VNet の再利用について

OpenShift Container Platform 4.5 では、クラスターを Microsoft Azure の既存の Azure Virtual Network (VNet) にデプロイできます。これを実行する場合、VNet 内の既存のサブネットおよびルーティングルールも使用する必要があります。

OpenShift Container Platform を既存の Azure VNet にデプロイすることで、新規アカウントでのサービス制限の制約を回避したり、会社のガイドラインによる運用上の制約をより容易に遵守することが可能になる場合があります。VNet の作成に必要なインフラストラクチャーの作成パーミッションを取得できない場合には、このオプションを使用できます。



重要

既存の VNet を使用するには、更新された Azure Private DNS (プレビュー) 機能を使用する必要があります。この機能の制限についての詳細は、[Announcing Preview Refresh for Azure DNS Private Zones](#) を参照してください。

1.6.3.1. VNet を使用するための要件

既存の VNet を使用してクラスターをデプロイする場合、クラスターをインストールする前に追加のネットワーク設定を実行する必要があります。インストーラーでプロビジョニングされるインフラストラクチャークラスターでは、インストーラーは通常以下のコンポーネントを作成しますが、既存の VNet にインストールする場合にはこれらを作成しません。

- サブネット
- ルートテーブル
- VNets
- ネットワークセキュリティグループ

カスタム VNet を使用する場合、インストールプログラムおよびクラスターでできるようにカスタム VNet およびそのサブネットを適切に設定する必要があります。インストールプログラムは、使用するクラスターのネットワーク範囲を細分化できず、サブネットのルートテーブルを設定するか、または DHCP などの VNet オプションを設定します。これは、クラスターのインストール前に設定する必要があります。

クラスターは、既存の VNet およびサブネットを含むリソースグループにアクセスできる必要があります。クラスターが作成するすべてのリソースは、作成される別個のリソースグループに配置され、一部のネットワークリソースが別個のグループから使用されます。一部のクラスター Operator は両方のリソースグループのリソースにアクセスできる必要があります。たとえば マシン API コントローラーは、ネットワークリソースグループから、作成される仮想マシンの NIC をサブネットに割り当てます。

VNet には以下の特徴が確認される必要があります。

- VNet の CIDR ブロックには、クラスターマシンの IP アドレスプールである **Networking.MachineCIDR** 範囲が含まれる必要があります。
- VNet およびそのサブネットは同じリソースグループに属する必要があり、サブネットは静的 IP アドレスではなく、Azure で割り当てられた DHCP IP アドレスを使用するように設定される必要があります。

コントロールプレーンマシンのサブネットおよびコンピューティングマシン用のサブネットの 2 つのサブネットを VNet 内に指定する必要があります。Azure はマシンを指定するリージョン内の複数の異なるアベイラビリティゾーンに分散するため、デフォルトのクラスターには高可用性があります。

指定するサブネットが適切であることを確認するには、インストールプログラムが以下のデータを確認します。

- 指定したサブネットすべてが存在します。

- コントロールプレーンマシンのサブネットおよびコンピューターマシンのサブネットの2つのサブネットを指定する必要があります。
- サブネットのCIDRは指定されたマシンCIDRに属します。マシンは、プライベートサブネットを指定しないアベイラビリティゾーンにはプロビジョニングされません。必要な場合に、インストールプログラムはコントロールプレーンおよびワーカーノードを管理するパブリックロードバランサーを作成し、AzureはパブリックIPアドレスをそれらに割り当てます。

既存のVNetを使用するクラスターを破棄しても、VNetは削除されません。

1.6.3.1.1. ネットワークセキュリティグループの要件

コンピューターマシンおよびコントロールプレーンマシンをホストするサブネットのネットワークセキュリティグループには、クラスターの通信が正しいことを確認するための特定のアクセスが必要です。必要なクラスター通信ポートへのアクセスを許可するルールを作成する必要があります。



重要

ネットワークセキュリティグループルールは、クラスターのインストール前に有効にされている必要があります。必要なアクセスなしにクラスターのインストールを試行しても、インストールプログラムはAzure APIに到達できず、インストールに失敗します。

表1.15 必須ポート

ポート	説明	コントロールプレーン	コンピューター
80	HTTP トラフィックを許可します。		x
443	HTTPS トラフィックを許可します		x
6443	コントロールプレーンマシンとの通信を許可します。	x	
22623	マシン設定サーバーとの通信を許可します。	x	



注記

クラスターコンポーネントは、Kubernetes コントローラーが更新する、ユーザーによって提供されるネットワークセキュリティグループを変更しないため、擬似セキュリティグループが環境の残りの部分に影響を及ぼさずに Kubernetes コントローラー用に作成されます。

1.6.3.2. パーミッションの区分

OpenShift Container Platform 4.3 以降、クラスターのデプロイに、インストールプログラムがプロビジョニングするインフラストラクチャクラスターに必要なすべてのパーミッションを必要としなくなりました。この変更は、ある会社で個人がクラウドで他とは異なるリソースを作成できるようにパーミッションが区分された状態に類似するものです。たとえば、インスタンス、ストレージ、ロードバランサーなどのアプリケーション固有のアイテムを作成することはできますが、VNet、サブネット、または Ingress ルールなどのネットワーク関連のコンポーネントは作成できない可能性があります。

クラスターの作成時に使用する Azure の認証情報には、VNet、およびサブネット、ルーティングテーブル、インターネットゲートウェイ、NAT、VPN などの VNet 内のコアとなるネットワークコンポーネントの作成に必要なネットワークのパーミッションは必要ありません。ロードバランサー、セキュリティグループ、ストレージアカウントおよびノードなどの、クラスター内でマシンに必要なアプリケーションリソースを作成するパーミッションは依然として必要になります。

1.6.3.3. クラスター間の分離

クラスターは既存のサブネットのネットワークセキュリティグループを変更できないため、VNet でクラスターを相互に分離する方法はありません。

1.6.4. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.6.5. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。

**注記**

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。

**注記**

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

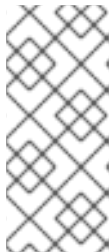
手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が `~/.ssh` ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。

**注記**

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ①
```

出力例

```
■
```

Identity added: /home/<you>/<path>/<file_name> (<computer_name>)

- 1 ~/.ssh/id_rsa などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.6.6. インストールプログラムの取得

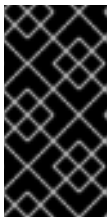
OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシー

クレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.6.7. インストール設定ファイルの手動作成

内部ネットワークからのみアクセスでき、インターネット上に表示されないプライベート OpenShift Container Platform クラスターのインストールの場合、インストール設定ファイルを手動で生成する必要があります。

前提条件

- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。

手順

1. 必要なインストールアセットを保存するためのインストールディレクトリを作成します。

```
$ mkdir <installation_directory>
```



重要

ディレクトリを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

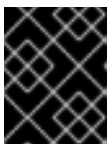
2. 以下の **install-config.yaml** ファイルテンプレートをカスタマイズし、これを **<installation_directory>** に保存します。



注記

この設定ファイル **install-config.yaml** に名前を付ける必要があります。

3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。



重要

install-config.yaml ファイルは、インストールプロセスの次の手順で使用されます。この時点でこれをバックアップする必要があります。

1.6.7.1. インストール設定パラメーター

OpenShift Container Platform クラスターをデプロイする前に、クラスターをホストするクラウドプラットフォームでアカウントを記述し、クラスターのプラットフォームをオプションでカスタマイズするためにパラメーターの値を指定します。**install-config.yaml** インストール設定ファイルを作成する際

に、コマンドラインで必要なパラメーターの値を指定します。クラスターをカスタマイズする場合、**install-config.yaml** ファイルを変更して、プラットフォームについての詳細情報を指定できます。



注記

インストール後は、これらのパラメーターを **install-config.yaml** ファイルで変更することはできません。



重要

openshift-install コマンドは、パラメーターのフィールド名を検証しません。正しくない名前を指定すると、関連するファイルまたはオブジェクトは作成されず、エラーが報告されません。指定されたパラメーターのフィールド名が正しいことを確認します。

1.6.7.1.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表1.16 必須パラメーター

パラメーター	説明	値
apiVersion	install-config.yaml コンテンツの API バージョン。現在のバージョンは v1 です。インストーラーは、古い API バージョンをサポートすることもできます。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、 baseDomain と <metadata.name> 、 <baseDomain> 形式を使用する metadata.name パラメーターの値の組み合わせです。	example.com などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース ObjectMeta 。ここからは name パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて {{.metadata.name}} 、 {{.baseDomain}} のサブドメインです。	dev などの小文字、ハイフン (-)、およびピリオド (.) が含まれる文字列。

パラメーター	説明	値
platform	インストールの実行に使用する特定プラットフォームの設定: aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。 platform.<platform> パラメーターに関する追加情報は、以下の表で特定のプラットフォームについて参照してください。	オブジェクト
pullSecret	https://cloud.redhat.com/openshift/install/pull-secret からプルシークレットを取得し、Quay.io などのサービスから OpenShift Container Platform コンポーネントのコンテナイメージのダウンロードを認証します。	<pre>{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }</pre>

1.6.7.1.2. ネットワーク設定パラメーター


既存のネットワークインフラストラクチャーの要件に基づいて、インストール設定をカスタマイズできます。たとえば、クラスターネットワークの IP アドレスブロックを拡張するか、デフォルトとは異なる IP アドレスブロックを指定できます。

IPv4 アドレスのみがサポートされます。

表1.17 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	<p>オブジェクト</p> <div>  <div> <p>注記</p> <p>インストール後に networking オブジェクトで指定したパラメーターを変更することはできません。</p> </div> </div>

パラメーター	説明	値
networking.networkType	インストールするクラスターネットワークプロバイダー Container Network Interface (CNI) プラグイン。	OpenShiftSDN または OVNKubernetes のいずれか。デフォルト値は OpenShiftSDN です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は 10.128.0.0/14 で、ホストの接頭辞は /23 です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	networking.clusterNetwork を使用する場合に必須です。IP アドレスブロック。 IPv4 ネットワーク	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は 0 から 32 の間になります。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、 hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます。 hostPrefix 値の 23 は、 $2^{(32 - 23) - 2}$ Pod IP アドレスを提供します。	サブネット接頭辞。 デフォルト値は 23 です。
networking.serviceNetwork	サービスの IP アドレスブロック。デフォルト値は 172.30.0.0/16 です。 OpenShift SDN および OVN-Kubernetes ネットワークプロバイダーは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。	CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。 networking: serviceNetwork: - 172.30.0.0/16
networking.machineNetwork	マシンの IP アドレスブロック。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: machineNetwork: - cidr: 10.0.0.0/16

パラメーター	説明	値
networking.machineNetwork.cidr	networking.machineNetwork を使用する場合に必須です。IP アドレスブロック。libvirt 以外のすべてのプラットフォームでは、デフォルト値は 10.0.0.0/16 です。libvirt の場合、デフォルト値は 192.168.126.0/24 です。	<p>CIDR 表記の IP ネットワークブロック。</p> <p>例: 10.0.0.0/16</p> <div>  <div> <p>注記</p> <p>優先される NIC が置かれている CIDR に一致する networking.machineNetwork を設定します。</p> </div> </div>

1.6.7.1.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表1.18 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシが設定される際にも使用できます。	文字列
compute	コンピュートノードを設定するマシンの設定。	machine-pool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列

パラメーター	説明	値
compute.hyperthreading	<p>コンピュータマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div>  <div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p> </div> </div>	Enabled または Disabled
compute.name	compute を使用する場合に必須です。マシンプールの名前。	worker
compute.platform	compute を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は controlPlane.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 o virt 、 vsphere 、または {}
compute.replicas	プロビジョニングするコンピュータマシン (ワーカーマシンとしても知られる) の数。	2 以上の正の整数。デフォルト値は 3 です。
controlPlane	コントロールプレーンを設定するマシンの設定。	MachinePool オブジェクトの配列。詳細は、以下の Machine-pool の表を参照してください。
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現時点で異種クラスターはサポートされていないため、すべてのプールが同じアーキテクチャーを指定する必要があります。有効な値は amd64 (デフォルト) です。	文字列

パラメーター	説明	値
controlPlane.hyperthreading	<p>コントロールプレーンマシンで同時マルチスレッドまたは hyperthreading を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。</p> <div>  <div> <p>重要</p> <p>同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れていることを確認します。</p> </div> </div>	Enabled または Disabled
controlPlane.name	controlPlane を使用する場合に必須です。マシンプールの名前。	master
controlPlane.platform	controlPlane を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は compute.platform パラメーターの値に一致する必要があります。	aws 、 azure 、 gcp 、 openstack 、 ovirt 、 vsphere 、または {}
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は 3 のみです (これはデフォルト値です)。
fips	<p>FIPS モードを有効または無効にします。デフォルトは false (無効) です。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。</p> <div>  <div> <p>注記</p> <p>Azure File ストレージを使用している場合、FIPS モードを有効にすることはできません。</p> </div> </div>	false または true

パラメーター	説明	値
imageContentSources	release-image コンテンツのソースおよびリポジトリ。	オブジェクトの配列。この表の以下の行で説明されているように、 source およびオプションで mirrors が含まれます。
imageContentSources.s.source	imageContentSources を使用する場合に必須です。ユーザーが参照するリポジトリを指定します (例: イメージプル仕様)。	文字列
imageContentSources.s.mirrors	同じイメージが含まれる可能性のあるリポジトリを1つ以上指定します。	文字列の配列。
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	Internal または External 。プライベートクラスターをデプロイするには、 publish を Internal に設定します。これはインターネットからアクセスできません。デフォルト値は External です。
sshKey	<p>クラスターマシンへのアクセスを認証するための SSH キー。</p> <div>  <p>注記</p> <p>インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、ssh-agent プロセスが使用する SSH キーを指定します。</p> </div>	たとえば、 sshKey: ssh-ed25519 AAAA.. です。

1.6.7.1.4. 追加の Azure 設定パラメーター

追加の Azure 設定パラメーターは以下の表で説明されています。

表1.19 追加の Azure パラメーター

パラメーター	説明	値
controlPlane.platform.azure.osDisk.diskSizeGB	VM の Azure ディスクのサイズ。	GB 単位でディスクのサイズを表す整数。サポートされる最小のディスクサイズは 1024 です。

パラメーター	説明	値
platform.azure.baseDomainResourceGroupName	ベースドメインの DNS ゾーンが含まれるリソースグループの名前。	文字列 (例: production_cluster)。
platform.azure.region	クラスターをホストする Azure リージョンの名前。	centralus などの有効なリージョン名。
platform.azure.zone	マシンを配置するアベイラビリティゾーンの一覧。高可用性を確保するには、少なくとも2つのゾーンを指定します。	ゾーンの一覧 (例: ["1", "2", "3"])。
platform.azure.networkResourceGroupName	クラスターをデプロイする既存の VNet を含むリソースグループの名前。この名前は platform.azure.baseDomainResourceGroupName と同じにすることはできません。	文字列。
platform.azure.virtualNetwork	クラスターをデプロイする既存 VNet の名前。	文字列。
platform.azure.controlPlaneSubnet	コントロールプレーンマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。
platform.azure.computeSubnet	コンピューターマシンをデプロイする VNet 内の既存サブネットの名前。	有効な CIDR (例: 10.0.0.0/16)。



注記

Azure クラスターで、[Azure アベイラビリティゾーン](#) のカスタマイズや [タグ](#) を使用した [Azure リソースの編成](#) を実行することはできません。

1.6.7.2. Azure のカスタマイズされた `install-config.yaml` ファイルのサンプル

`install-config.yaml` ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。



重要

このサンプルの YAML ファイルは参照用にのみ提供されます。インストールプログラムを使用して `install-config.yaml` ファイルを取得し、これを変更する必要があります。

apiVersion: v1

baseDomain: example.com **1**

```

controlPlane: 2
  hyperthreading: Enabled 3 4
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 5
        type: Standard_D8s_v3
      replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
    replicas: 5
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
    networkResourceGroupName: vnet_resource_group 13
    virtualNetwork: vnet 14
    controlPlaneSubnet: control_plane_subnet 15
    computeSubnet: compute_subnet 16
  pullSecret: '{"auths": ...}' 17
  fips: false 18
  sshKey: ssh-ed25519 AAAA... 19
  publish: Internal 20

```

1 10 11 17 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

2 6 これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

3 7 **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン - で始め、**controlPlane** セクションの最初の行はハイフンで始めることができませ

ん。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1つのコントロールプレーンプールのみが使用されます。

- 4 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。



重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。同時マルチスレッドを無効にする場合は、マシンに対して **Standard_D8s_v3** などの大規模な仮想マシンタイプを使用します。

- 5 8 使用するディスクのサイズは、GB 単位で指定できます。マスターノードの最小推奨値は 1024 GB です。
- 9 マシンをデプロイするゾーンの一覧を指定します。高可用性を確保するには、少なくとも 2 つのゾーンを指定します。
- 12 ベースドメインの DNS ゾーンが含まれるリソースグループの名前を指定します。
- 13 既存の VNet を使用する場合は、それが含まれるリソースグループの名前を指定します。
- 14 既存の VNet を使用する場合は、その名前を指定します。
- 15 既存の VNet を使用する場合は、コントロールプレーンマシンをホストするサブネットの名前を指定します。
- 16 既存の VNet を使用する場合は、コンピュートマシンをホストするサブネットの名前を指定します。
- 18 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。
- 19 クラスター内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

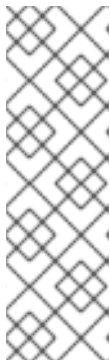
- 20 クラスターのユーザーに表示されるエンドポイントをパブリッシュする方法。プライベートクラスターをデプロイするには、**publish** を **Internal** に設定します。これはインターネットからアクセスできません。デフォルト値は **External** です。

1.6.7.3. インストール時のクラスター全体のプロキシの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシを使用することができます。プロキシ設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスターをプロキシを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイルが必要です。
- クラスターがアクセスする必要のあるサイトを確認し、プロキシをバイパスする必要があるかどうかを判別します。デフォルトで、すべてのクラスター egress トラフィック (クラスターをホストするクラウドについてのクラウドプロバイダー API に対する呼び出しを含む) はプロキシされます。**Proxy** オブジェクトの **spec.noProxy** フィールドにサイトを追加し、必要に応じてプロキシをバイパスします。



注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud Platform (GCP)、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP) へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント (**169.254.169.254**) も設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシ設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: http://<username>:<pswd>@<ip>:<port> ❷
  noProxy: example.com ❸
additionalTrustBundle: | ❹
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----
...
```

- ❶ クラスター外の HTTP 接続を作成するために使用するプロキシ URL。URL スキームは **http** である必要があります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpProxy** 値を指定することはできません。
- ❷ クラスター外で HTTPS 接続を作成するために使用するプロキシ URL。このフィールドが指定されていない場合、HTTP および HTTPS 接続の両方に **httpProxy** が使用されます。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。
- ❸ プロキシを除外するための宛先ドメイン名、ドメイン、IP アドレス、または他のネットワーク CIDR のコンマ区切りの一覧。サブドメインのみと一致するように、ドメインの前に **.** を付けます。たとえば、**.y.com** は **x.y.com** に一致しますが、**y.com** には一致しません。

ん。*を使用し、すべての宛先のプロキシをバイパスします。

- 4 指定されている場合、インストールプログラムは HTTPS 接続のプロキシに必要な1つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシのアイデンティティ証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。



注記

インストールプログラムは、プロキシの **readinessEndpoints** フィールドをサポートしません。

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシ設定を使用する **cluster** という名前のクラスター全体のプロキシを作成します。プロキシ設定が指定されていない場合、**cluster Proxy** オブジェクトが依然として作成されますが、これには **spec** がありません。



注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシを作成することはできません。

1.6.8. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に1回だけ実行できます。

前提条件

- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

```
$ ./openshift-install create cluster --dir=<installation_directory> \ 1
--log-level=info 2
```

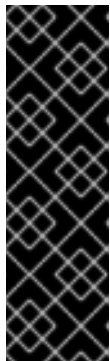
- 1 **<installation_directory>** については、以下を指定します。
- 2 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



注記

ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

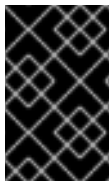


重要

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

1.6.9. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。



重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.6.9.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。

3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。**PATH** を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.6.9.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。**PATH** を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.6.9.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。

3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.6.10. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。
- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

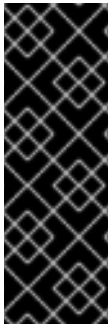
1.6.11. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

1.7. ARM テンプレートを使用したクラスターの AZURE へのインストール

OpenShift Container Platform バージョン 4.5 では、独自にプロビジョニングするインフラストラクチャーを使用して、クラスターを Microsoft Azure にインストールできます。

これらの手順を実行するか、独自の手順を作成するのに役立つ複数の [Azure Resource Manager \(ARM\)](#) テンプレートが提供されます。

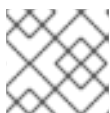


重要

ユーザーによってプロビジョニングされるインフラストラクチャーのインストールする手順は、例としてのみ提供されます。独自にプロビジョニングするインフラストラクチャーでクラスターをインストールするには、クラウドプロバイダーおよび OpenShift Container Platform のインストールプロセスについて理解する必要があります。これらの手順を実行するか、独自の手順を作成するのに役立つ複数の ARM テンプレートが提供されます。他の方法を使用して必要なリソースを作成することもできます。これらのテンプレートはサンプルとしてのみ提供されます。

1.7.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- [Azure アカウントを設定](#) してクラスターをホストします。
- Azure CLI をダウンロードし、これをコンピューターにインストールします。Azure ドキュメントの [Install the Azure CLI](#) を参照してください。以下のドキュメントについては、直近で Azure CLI のバージョン **2.2.0** を使用してテストされています。Azure CLI コマンドは、使用するバージョンによって動作が異なる場合があります。
- ファイアウォールを使用し、Telemetry を使用する予定がある場合は、クラスターがアクセスする必要のある [サイトを許可するようにファイアウォールを設定](#) する必要があります。
- システムが IAM(アイデンティティおよびアクセス管理) を管理できない場合、クラスター管理者は [IAM 認証情報を手動で作成し、維持](#) できます。手動モードは、クラウド IAM API に到達できない環境でも使用できます。



注記

プロキシを設定する場合は、このサイトを閲覧も確認してください。

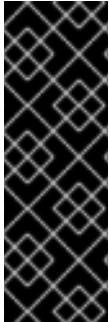
1.7.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.5 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、[subscription watch](#) を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

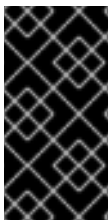


重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.7.3. Azure プロジェクトの設定

OpenShift Container Platform をインストールする前に、これをホストするために Azure プロジェクトを設定する必要があります。



重要

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

1.7.3.1. Azure アカウントの制限

OpenShift Container Platform クラスターは数多くの Microsoft Azure コンポーネントを使用し、デフォルトの [Azure サブスクリプションおよびサービス制限、クォータ、および制約](#) は、OpenShift Container Platform クラスターをインストールする機能に影響を与えます。



重要

デフォルトの制限は、Free Trial や Pay-As-You-Go、および DV2、F、および G などのシリーズといったカテゴリータイプによって異なります。たとえば、Enterprise Agreement サブスクリプションのデフォルトは 350 コアです。

サブスクリプションタイプの制限を確認し、必要に応じて、デフォルトのクラスターを Azure にインストールする前にアカウントのクォータ制限を引き上げます。

以下の表は、OpenShift Container Platform クラスターのインストールおよび実行機能に影響を与える可能性のある Azure コンポーネントの制限を要約しています。

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明
vCPU	40	リージョンごとに 20	<p>デフォルトのクラスターには 40 の vCPU が必要であるため、アカウントの上限を引き上げる必要があります。</p> <p>デフォルトで、各クラスターは以下のインスタンスを作成します。</p> <ul style="list-style-type: none"> ● 1つのブートストラップマシン。これはインストール後に削除されます。 ● 3つのコントロールプレーンマシン ● 3つのコンピュートマシン <p>ブートストラップマシンは 4 vCPUS を使用する Standard_D4s_v3 マシンを使用し、コントロールプレーンマシンは 8 vCPU を使用する Standard_D8s_v3 仮想マシンを使用し、さらにワーカーマシンは、4 vCPU を使用する Standard_D4s_v3 仮想マシンを使用するため、デフォルトクラスターには 40 の vCPU が必要になります。4 vCPU を使用するブートストラップノードの仮想マシンは、インストール時にのみ使用されます。</p> <p>追加のワーカーノードをデプロイし、自動スケーリングを有効にし、大規模なワークロードをデプロイするか、または異なるインスタンスタイプを使用するには、アカウントの vCPU 制限をさらに引き上げ、クラスターが必要なマシンをデプロイできるようにする必要があります。</p> <p>デフォルトで、インストールプログラムはコントロールプレーンおよびコンピュートマシンを、リージョン内のすべてのアベイラビリティゾーン に分散します。クラスターの高可用性を確保するには、少なくとも 3 つ以上のアベイラビリティゾーンのあるリージョンを選択します。リージョンに含まれるアベイラビリティゾーンが 3 つ未満の場合、インストールプログラムは複数のコントロールプレーンマシンを利用可能なゾーンに配置します。</p>
VNet	1	リージョンごとに 1000	各デフォルトクラスターには、2 つのサブネットを含む 1 つの Virtual Network (VNet) が必要です。
ネットワークインターフェイス	6	リージョンごとに 65,536	各デフォルトクラスターには、6 つのネットワークインターフェイスが必要です。さらに多くのマシンを作成したり、デプロイしたワークロードでロードバランサーを作成する場合、クラスターは追加のネットワークインターフェイスを使用します。

コンポーネント	デフォルトで必要なコンポーネントの数	デフォルトの Azure 制限	説明						
ネットワークセキュリティグループ	2	5000	<div>各デフォルトクラスター。各クラスターは VNet の各サブネットにネットワークセキュリティグループを作成します。デフォルトのクラスターは、コントロールプレーンおよびコンピューターノードのサブネットにネットワークセキュリティグループを作成します。</div> <table><tr><td>control plane</td><td>任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。</td></tr><tr><td>node</td><td>インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。</td></tr></table>	control plane	任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。	node	インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。		
control plane	任意の場所からコントロールプレーンマシンにポート 6443 でアクセスできるようにします。								
node	インターネットからワーカーノードにポート 80 および 443 でアクセスできるようにします。								
ネットワークロードバランサー	3	リージョンごとに 1000	<div>各クラスターは以下の ロードバランサー を作成します。</div> <table><tr><td>default</td><td>ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス</td></tr><tr><td>internal</td><td>コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス</td></tr><tr><td>external</td><td>コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス</td></tr></table> <div>アプリケーションが追加の Kubernetes LoadBalancer サービスオブジェクトを作成すると、クラスターは追加のロードバランサーを使用します。</div>	default	ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス	internal	コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス	external	コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス
default	ワーカーマシン間でポート 80 および 443 での要求の負荷分散を行うパブリック IP アドレス								
internal	コントロールプレーンマシン間でポート 6443 および 22623 での要求の負荷分散を行うプライベート IP アドレス								
external	コントロールプレーンマシン間でポート 6443 での要求の負荷分散を行うパブリック IP アドレス								
パブリック IP アドレス	3		2 つのパブリックロードバランサーのそれぞれはパブリック IP アドレスを使用します。ブートストラップマシンは、インストール時のトラブルシューティングのためにマシンに SSH を実行できるようにパブリック IP アドレスも使用します。ブートストラップノードの IP アドレスは、インストール時にのみ使用されます。						
プライベート IP アドレス	7		内部ロードバランサー、3 つのコントロールプレーンマシンのそれぞれ、および 3 つのワーカーマシンのそれぞれはプライベート IP アドレスを使用します。						

1.7.3.2. Azure でのパブリック DNS ゾーンの設定

OpenShift Container Platform をインストールするには、使用する Microsoft Azure アカウントに、専用のパブリックホスト DNS ゾーンが必要になります。このゾーンはドメインに対する権威を持っている必要があります。このサービスは、クラスターへの外部接続のためのクラスター DNS 解決および名前検索を提供します。

手順

1. ドメイン、またはサブドメイン、およびレジストラを特定します。既存のドメインおよびレジストラを移行するか、Azure または別のソースから新規のものを取得できます。



注記

Azure 経由でドメインを購入する方法についての詳細は、Azure ドキュメントの [Buy a custom domain name for Azure App Service](#) を参照してください。

2. 既存のドメインおよびレジストラを使用している場合、その DNS を Azure に移行します。Azure ドキュメントの [Migrate an active DNS name to Azure App Service](#) を参照してください。
3. ドメインの DNS を設定します。Azure ドキュメントの [Tutorial: Host your domain in Azure DNS](#) の手順に従い、ドメインまたはサブドメインのパブリックホストゾーンを作成し、新規の権威ネームサーバーを抽出し、ドメインが使用するネームサーバーのレジストラレコードを更新します。
openshiftcorp.com などのルートドメインや、**clusters.openshiftcorp.com** などのサブドメインを使用します。
4. サブドメインを使用する場合は、所属する会社の手順に従ってその委任レコードを親ドメインに追加します。

この [DNS ゾーン](#) の作成例 を参照し、Azure の DNS ソリューションを確認することができます。

1.7.3.3. Azure アカウント制限の拡張

アカウントの制限を引き上げるには、Azure ポータルでサポートをリクエストします。



注記

サポートリクエストごとに1つの種類のクォータのみを増やすことができます。

手順

1. Azure ポータルの左端で **Help + support** をクリックします。
2. **New support request** をクリックしてから必要な値を選択します。
 - a. **Issue type** 一覧から、**Service and subscription limits (quotas)** を選択します。
 - b. **Subscription** 一覧から、変更するサブスクリプションを選択します。
 - c. **Quota type** 一覧から、引き上げるクォータを選択します。たとえば、**Compute-VM (cores-vCPUs) subscription limit increases** を選択し、クラスターのインストールに必要な vCPU の数を増やします。

- d. **Next: Solutions** をクリックします。
3. **Problem Details** ページで、クォータの引き上げについての必要な情報を指定します。
 - a. **Provide details** をクリックし、**Quota details** ウィンドウに必要な詳細情報を指定します。
 - b. **SUPPORT METHOD and CONTACT INFO** セクションに、問題の重大度および問い合わせ先の詳細を指定します。
4. **Next: Review + create** をクリックしてから **Create** をクリックします。

1.7.3.4. 証明書署名要求の管理

ユーザーがプロビジョニングするインフラストラクチャーを使用する場合、クラスターの自動マシン管理へのアクセスは制限されるため、インストール後にクラスターの証明書署名要求 (CSR) のメカニズムを提供する必要があります。**kube-controller-manager** は kubelet クライアント CSR のみを承認します。**machine-approver** は、kubelet 認証情報を使用して要求される提供証明書の有効性を保証できません。適切なマシンがこの要求を発行したかどうかを確認できないためです。kubelet 提供証明書の要求の有効性を検証し、それらを承認する方法を判別し、実装する必要があります。

1.7.3.5. 必要な Azure ロール

Microsoft Azure アカウントには、使用するサブスクリプションについて以下のロールが必要です。

- **User Access Administrator**

Azure ポータルでロールを設定するには、Azure ドキュメントの [Manage access to Azure resources using RBAC and the Azure portal](#) を参照します。

1.7.3.6. サービスプリンシパルの作成

OpenShift Container Platform およびそのインストールプログラムは Azure Resource Manager 経由で Microsoft Azure リソースを作成する必要があるため、これを表すサービスプリンシパルを作成する必要があります。

前提条件

- [Azure CLI](#) のインストールまたは更新を実行します。
- **jq** パッケージをインストールします。
- Azure アカウントには、使用するサブスクリプションに必要なロールがなければなりません。

手順

1. Azure CLI にログインします。

```
$ az login
```

認証情報を使用して Web コンソールで Azure にログインします。

2. Azure アカウントでサブスクリプションを使用している場合は、適切なサブスクリプションを使用していることを確認してください。
 - a. 利用可能なアカウントの一覧を表示し、クラスターに使用するサブスクリプションの **tenantId** の値を記録します。

```
$ az account list --refresh
```

出力例

```
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

- b. アクティブなアカウントの詳細を表示し、**tenantId** 値が使用するサブスクリプションと一致することを確認します。

```
$ az account show
```

出力例

```
{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", ❶
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

- ❶ **tenantId** パラメーターの値が適切なサブスクリプションの UUID であることを確認します。

- c. 適切なサブスクリプションを使用していない場合には、アクティブなサブスクリプションを変更します。

```
$ az account set -s <id> ❶
```

- ❶ 使用する必要のあるサブスクリプションの **id** の値を **<id>** の代わりに使用します。

- d. アクティブなサブスクリプションを変更したら、アカウント情報を再度表示します。

```
$ az account show
```

出力例

```
{
  "environmentName": "AzureCloud",
  "id": "33212d16-bdf6-45cb-b038-f6565b61edda",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

3. 直前の出力の **tenantId** および **id** パラメーターの値を記録します。OpenShift Container Platform のインストール時にこれらの値が必要になります。
4. アカウントのサービスプリンシパルを作成します。

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> ❶
```

- ❶ **<service_principal>** を、サービスプリンシパルに割り当てる名前に置き換えます。

出力例

```
Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}
```

5. 直前の出力の **appId** および **password** パラメーターの値を記録します。OpenShift Container Platform のインストール時にこれらの値が必要になります。
6. サービスプリンシパルに追加パーミッションを付与します。サービスプリンシパルには、クラスターでそのコンポーネントの認証情報を割り当てられるようにレガシーの **Azure Active Directory Graph** → **Application.ReadWrite.OwnedBy** パーミッションおよび **User Access Administrator** ロールが必要です。
 - a. **User Access Administrator** ロールを割り当てるには、以下のコマンドを実行します。

```
$ az role assignment create --role "User Access Administrator" \
  --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \
    | jq '[0].objectId' -r) ❶
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

- b. **Azure Active Directory Graph** パーミッションを割り当てるには、以下のコマンドを実行します。

```
$ az ad app permission add --id <appId> \ ❶
  --api 00000002-0000-0000-c000-000000000000 \
  --api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

出力例

```
Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --api
00000002-0000-0000-c000-000000000000" is needed to make the change effective
```

このコマンドで付与する特定のパーミッションについての詳細は、[GUID Table for Windows Azure Active Directory Permissions](#) を参照してください。

- c. パーミッション要求を承認します。アカウントに Azure Active Directory テナント管理者ロールがない場合は、所属する組織のガイドラインに従い、テナント管理者にパーミッション要求を承認するようにリクエストしてください。

```
$ az ad app permission grant --id <appId> \ ❶
  --api 00000002-0000-0000-c000-000000000000
```

❶ <appId> を、サービスプリンシパルの **appId** パラメーター値に置き換えます。

1.7.3.7. サポート対象の Azure リージョン

インストールプログラムは、サブスクリプションに基づいて利用可能な Microsoft Azure リージョンの一覧を動的に生成します。以下の Azure リージョンは OpenShift Container Platform バージョン 4.5.4 でテストされ、検証されています。

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)

- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)
- **eastus2** (East US 2)
- **francecentral** (France Central)
- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)
- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)
- **norwayeast** (Norway East)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

1.7.4. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

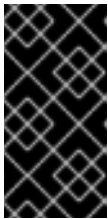
手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

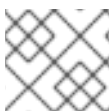
3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.7.5. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。このキーを使用してパブリッククラスターのブートストラップマシンにアクセスし、インストールの問題をトラブルシューティングできます。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、新規 SSH キーのパスおよびファイル名を指定します。既存のキーペアがある場合は、公開鍵が `~/.ssh` ディレクトリーにあることを確認します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。



注記

FIPS で検証済み/進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

1. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> ①
```

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。クラスターを独自にプロビジョニングするインフラストラクチャーにインストールする場合は、このキーをクラスターのマシンに指定する必要があります。

1.7.6. AWS のインストールファイルの作成

ユーザーによってプロビジョニングされるインフラストラクチャーを使用して OpenShift Container Platform を Microsoft Azure にインストールするには、インストールプログラムがクラスターをデプロイするために必要なファイルを生成し、クラスターが使用するマシンのみを作成するようにそれらのファイルを変更する必要があります。**install-config.yaml** ファイル、Kubernetes マニフェスト、および Ignition 設定ファイルを生成し、カスタマイズします。

1.7.6.1. インストール設定ファイルの作成

Microsoft Azure にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. **install-config.yaml** ファイルを作成します。

- a. 以下のコマンドを実行します。

```
$ ./openshift-install create install-config --dir=<installation_directory> ❶
```

- ❶ **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。



重要

空のディレクトリーを指定します。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

- b. プロンプト時に、クラウドの設定の詳細情報を指定します。
 - i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。

**注記**

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

- ii. ターゲットに設定するプラットフォームとして **azure** を選択します。
- iii. お使いのコンピューターに Microsoft Azure プロファイルが保存されていない場合は、サブスクリプションとサービスプリンシパルに以下の Azure パラメーター値を指定します。
 - **azure subscription id** クラスターに使用するサブスクリプション ID。アカウント出力に **id** 値を指定します。
 - **azure tenant id** テナント ID。アカウント出力に **tenantId** 値を指定します。
 - **azure service principal client id** サービスプリンシパルの **appId** パラメーターの値。
 - **azure service principal client secret** サービスプリンシパルの **password** パラメーターの値。
- iv. クラスターをデプロイするリージョンを選択します。
- v. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成した Azure DNS ゾーンに対応します。
- vi. クラスターの記述名を入力します。

**重要**

パブリックエンドポイントで利用可能なすべての Azure リソースはリソース名の制限を受けるため、特定の用語を使用するリソースを作成することはできません。Azure が制限する語の一覧は、Azure ドキュメントの [Resolve reserved resource name errors](#) を参照してください。

- vii. Red Hat OpenShift Cluster Manager サイトの [Pull Secret](#) ページから取得したプルシークレットを貼り付けます。
2. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細については、[インストール設定パラメーターセクション](#)を参照してください。
3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。

**重要**

install-config.yaml ファイルはインストールプロセス時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

1.7.6.2. インストール時のクラスター全体のプロキシの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシを使用することができます。プロキシ設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスタをプロキシを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイルが必要です。
- クラスタがアクセスする必要のあるサイトを確認し、プロキシをバイパスする必要があるかどうかを判別します。デフォルトで、すべてのクラスタ egress トラフィック (クラスタをホストするクラウドについてのクラウドプロバイダー API に対する呼び出しを含む) はプロキシされます。**Proxy** オブジェクトの **spec.noProxy** フィールドにサイトを追加し、必要に応じてプロキシをバイパスします。



注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud Platform (GCP)、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP) へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント (**169.254.169.254**) も設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシ設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> ❶
  httpsProxy: http://<username>:<pswd>@<ip>:<port> ❷
  noProxy: example.com ❸
additionalTrustBundle: | ❹
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----
...
```

- ❶ クラスタ外の HTTP 接続を作成するために使用するプロキシ URL。URL スキームは **http** である必要があります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpProxy** 値を指定することはできません。
- ❷ クラスタ外で HTTPS 接続を作成するために使用するプロキシ URL。このフィールドが指定されていない場合、HTTP および HTTPS 接続の両方に **httpProxy** が使用されます。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。
- ❸ プロキシを除外するための宛先ドメイン名、ドメイン、IP アドレス、または他のネットワーク CIDR のコンマ区切りの一覧。サブドメインのみと一致するように、ドメインの前に **.** を付けます。たとえば、**.y.com** は **x.y.com** に一致しますが、**y.com** には一致しません。

ん。*を使用し、すべての宛先のプロキシをバイパスします。

- 4 指定されている場合、インストールプログラムは HTTPS 接続のプロキシに必要な1つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシのアイデンティティ証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。



注記

インストールプログラムは、プロキシの **readinessEndpoints** フィールドをサポートしません。

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシ設定を使用する **cluster** という名前のクラスター全体のプロキシを作成します。プロキシ設定が指定されていない場合、**cluster Proxy** オブジェクトが依然として作成されますが、これには **spec** がありません。



注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシを作成することはできません。

1.7.6.3. ARM テンプレートの一般的な変数のエクスポート

ユーザーによって提供されるインフラストラクチャーのインストールを Microsoft Azure で実行するのに役立つ指定の Azure Resource Manager (ARM) テンプレートで使用される一般的な変数のセットをエクスポートする必要があります。



注記

特定の ARM テンプレートには、追加のエクスポートされる変数が必要になる場合があります。これについては、関連する手順で詳しく説明されています。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. 提供される ARM テンプレートで使用される **install-config.yaml** にある一般的な変数をエクスポートします。

```
$ export CLUSTER_NAME=<cluster_name> 1
$ export AZURE_REGION=<azure_region> 2
$ export SSH_KEY=<ssh_key> 3
```

```
$ export BASE_DOMAIN=<base_domain> 4
$ export BASE_DOMAIN_RESOURCE_GROUP=<base_domain_resource_group> 5
```

- 1 **install-config.yaml** ファイルからの **.metadata.name** 属性の値。
- 2 クラスタをデプロイするリージョン (例: **centralus**)。これは、**install-config.yaml** ファイルからの **.platform.azure.region** 属性の値です。
- 3 文字列としての SSH RSA 公開鍵ファイル。SSH キーは、スペースが含まれているために引用符で囲む必要があります。これは、**install-config.yaml** ファイルからの **.sshKey** 属性の値です。
- 4 クラスタをデプロイするベースドメイン。ベースドメインは、クラスタに作成したパブリック DNS ゾーンに対応します。これは、**install-config.yaml** からの **.baseDomain** 属性の値です。
- 5 パブリック DNS ゾーンが存在するリソースグループ。これは、**install-config.yaml** ファイルからの **.platform.azure.baseDomainResourceGroupName** 属性の値です。

以下に例を示します。

```
$ export CLUSTER_NAME=test-cluster
$ export AZURE_REGION=centralus
$ export SSH_KEY="ssh-rsa xxx/xxx/xxx= user@email.com"
$ export BASE_DOMAIN=example.com
$ export BASE_DOMAIN_RESOURCE_GROUP=ocp-cluster
```

2. kubeadmin 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

1.7.6.4. Kubernetes マニフェストおよび Ignition 設定ファイルの作成

一部のクラスタ定義ファイルを変更し、クラスタマシンを手動で起動する必要があるため、クラスタがマシンを作成するために必要な Kubernetes マニフェストと Ignition 設定ファイルを生成する必要があります。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスタが停止し、24 時間経過した後にクラスタを再起動すると、クラスタは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

前提条件

- OpenShift Container Platform インストールプログラムを取得します。

- **install-config.yaml** インストール設定ファイルを作成します。

手順

1. クラスターの Kubernetes マニフェストを生成します。

```
$ ./openshift-install create manifests --dir=<installation_directory> 1
```

出力例

```
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
```

- 1** **<installation_directory>** については、作成した **install-config.yaml** ファイルが含まれるインストールディレクトリーを指定します。

インストールプロセスの後の部分で独自のコンピュータマシンを作成するため、この警告を無視しても問題がありません。

2. コントロールプレーンマシンを定義する Kubernetes マニフェストファイルを削除します。

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_master-machines-*.yaml
```

これらのファイルを削除することで、クラスターがコントロールプレーンマシンを自動的に生成するのを防ぐことができます。

3. ワーカーマシンを定義する Kubernetes マニフェストファイルを削除します。

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

ワーカーマシンは独自に作成し、管理するため、これらのマシンを初期化する必要はありません。

4. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。
 - a. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** ファイルを開きます。
 - b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。
 - c. ファイルを保存し、終了します。
5. オプション: [Ingress Operator](#) を DNS レコードを作成するよう設定する必要がない場合は、**<installation_directory>/manifests/cluster-dns-02-config.yml** DNS 設定ファイルから **privateZone** および **publicZone** セクションを削除します。

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
```

```
spec:
  baseDomain: example.openshift.com
  privateZone: ❶
    id: mycluster-100419-private-zone
  publicZone: ❷
    id: example.openshift.com
  status: {}
```

- ❶ ❷ このセクションを完全に削除します。

これを実行する場合、後のステップで Ingress DNS レコードを手動で追加する必要があります。

6. ユーザーによってプロビジョニングされるインフラストラクチャーで Azure を設定する場合、Azure Resource Manager (ARM) テンプレートで後に使用するためにマニフェストファイルに定義された一般的な変数の一部をエクスポートする必要があります。
 - a. 以下のコマンドを使用してインフラストラクチャー ID をエクスポートします。

```
$ export INFRA_ID=<infra_id> ❶
```

- ❶ OpenShift Container Platform クラスターには、**<cluster_name>-<random_string>** の形式の識別子 (**INFRA_ID**) が割り当てられます。これは、提供される ARM テンプレートを使用して作成されるほとんどのリソースのベース名として使用されます。これは、**manifests/cluster-infrastructure-02-config.yml** ファイルからの **.status.infrastructureName** 属性の値です。

- b. 以下のコマンドを使用してリソースグループをエクスポートします。

```
$ export RESOURCE_GROUP=<resource_group> ❶
```

- ❶ この Azure デプロイメントで作成されたすべてのリソースは、**リソースグループ**の一部として存在します。リソースグループ名は、**<cluster_name>-<random_string>-rg** 形式の **INFRA_ID** をベースとしています。これは、**manifests/cluster-infrastructure-02-config.yml** ファイルからの **.status.platformStatus.azure.resourceGroupName** 属性の値です。

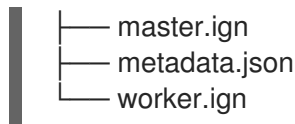
7. Ignition 設定ファイルを取得します。

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> ❶
```

- ❶ **<installation_directory>** については、同じインストールディレクトリーを指定します。

以下のファイルはディレクトリーに生成されます。

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
└── bootstrap.ign
```



1.7.7. Azure リソースグループおよびアイデンティティの作成

Microsoft Azure [リソースグループ](#) およびリソースグループのアイデンティティを作成する必要があります。これらはいずれも Azure での OpenShift Container Platform クラスターのインストール時に使用されます。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

手順

1. サポートされる Azure リージョンにリソースグループを作成します。

```
$ az group create --name ${RESOURCE_GROUP} --location ${AZURE_REGION}
```

2. リソースグループの Azure アイデンティティを作成します。

```
$ az identity create -g ${RESOURCE_GROUP} -n ${INFRA_ID}-identity
```

これは、クラスター内の Operator に必要なアクセスを付与するために使用されます。たとえば、これにより Ingress Operator はパブリック IP およびそのロードバランサーを作成できます。Azure アイデンティティをロールに割り当てる必要があります。

3. Contributor ロールを Azure アイデンティティに付与します。

- a. Azure ロールの割り当てで必要な以下の変数をエクスポートします。

```
$ export PRINCIPAL_ID=`az identity show -g ${RESOURCE_GROUP} -n ${INFRA_ID}-identity --query principalId --out tsv`
```

```
$ export RESOURCE_GROUP_ID=`az group show -g ${RESOURCE_GROUP} --query id --out tsv`
```

- b. Contributor ロールをアイデンティティに割り当てます。

```
$ az role assignment create --assignee "${PRINCIPAL_ID}" --role 'Contributor' --scope "${RESOURCE_GROUP_ID}"
```

1.7.8. RHCOS クラスターイメージおよびブートストラップ Ignition 設定ファイルのアップロード

Azure クライアントは、ローカルにあるファイルに基づくデプロイメントをサポートしないため、RHCOS 仮想ハードディスク (VHD) クラスターイメージおよびブートストラップ Ignition 設定ファイルをコピーし、それらをストレージコンテナに保存し、それらをデプロイメント時にアクセスできるようにする必要があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

手順

1. VHD クラスターイメージを保存するために Azure ストレージアカウントを作成します。

```
$ az storage account create -g ${RESOURCE_GROUP} --location ${AZURE_REGION} --name ${CLUSTER_NAME}sa --kind Storage --sku Standard_LRS
```



警告

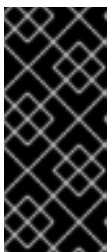
Azure ストレージアカウント名は 3 文字から 24 文字の長さで、数字および小文字のみを使用する必要があります。**CLUSTER_NAME** 変数がこれらの制限に準拠しない場合、Azure ストレージアカウント名を手動で定義する必要があります。Azure ストレージアカウント名の制限についての詳細は、Azure ドキュメントの [Resolve errors for storage account names](#) を参照してください。

2. ストレージアカウントキーを環境変数としてエクスポートします。

```
$ export ACCOUNT_KEY=`az storage account keys list -g ${RESOURCE_GROUP} --account-name ${CLUSTER_NAME}sa --query "[0].value" -o tsv`
```

3. 使用する RHCOS バージョンを選択し、その VHD の URL を環境変数にエクスポートします。

```
$ export VHD_URL=`curl -s https://raw.githubusercontent.com/openshift/installer/release-4.5/data/data/rhcos.json | jq -r .azure.url`
```



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージを指定する必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。

4. 選択した VHD を blob にコピーします。

```
$ az storage container create --name vhd --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY}
```

```
$ az storage blob copy start --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} --destination-blob "rhcos.vhd" --destination-container vhd --source-uri "${VHD_URL}"
```

VHD コピータスクの進捗を追跡するには、以下のスクリプトを実行します。

```
status="unknown"
while [ "$status" != "success" ]
do
  status=`az storage blob show --container-name vhd --name "rhcos.vhd" --account-name
${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} -o tsv --query
properties.copy.status`
  echo $status
done
```

- blob ストレージコンテナを作成し、生成された **bootstrap.ign** ファイルをアップロードします。

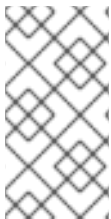
```
$ az storage container create --name files --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} --public-access blob
```

```
$ az storage blob upload --account-name ${CLUSTER_NAME}sa --account-key
${ACCOUNT_KEY} -c "files" -f "<installation_directory>/bootstrap.ign" -n "bootstrap.ign"
```

1.7.9. DNS ゾーンの作成例

DNS レコードは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用するクラスターに必要です。シナリオに適した DNS ストラテジーを選択する必要があります。

この例の場合、[Azure の DNS ソリューション](#) が使用されるため、外部 (インターネット) の可視性のために新規パブリック DNS ゾーンと、内部クラスターの解決用にプライベート DNS ゾーンが作成されます。



注記

パブリック DNS ゾーンは、クラスターデプロイメントと同じリソースグループに存在している必要はなく、必要なベースドメイン用にすでに組織内に存在している可能性があります。その場合、パブリック DNS ゾーンの作成を省略できます。先に生成したインストール設定がこのシナリオに基づいていることを確認してください。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

手順

- BASE_DOMAIN_RESOURCE_GROUP** 環境変数でエクスポートされたリソースグループに、新規のパブリック DNS ゾーンを作成します。

```
$ az network dns zone create -g ${BASE_DOMAIN_RESOURCE_GROUP} -n
${CLUSTER_NAME}.${BASE_DOMAIN}
```

すでに存在するパブリック DNS ゾーンを使用している場合は、この手順を省略できます。

2. このデプロイメントの残りの部分と同じリソースグループにプライベート DNS ゾーンを作成します。

```
$ az network private-dns zone create -g ${RESOURCE_GROUP} -n  
${CLUSTER_NAME}.${BASE_DOMAIN}
```

[Azure でのパブリック DNS ゾーンの設定](#) についてのセクションを参照してください。

1.7.10. Azure での VNet の作成

OpenShift Container Platform クラスター用に Microsoft Azure で使用する仮想ネットワーク (VNet) を作成する必要があります。各種の要件を満たすように VPC をカスタマイズできます。VNet を作成する方法として、提供される Azure Resource Manager (ARM) テンプレートを変更することができます。



注記

提供される ARM テンプレートを使用して Azure インフラストラクチャーを使用しない場合、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせる必要がある可能性があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

手順

1. 本トピックの **VNet の ARM テンプレート** セクションからテンプレートをコピーし、これを **01_vnet.json** としてクラスターのインストールディレクトリーに保存します。このテンプレートは、クラスターに必要な VNet について記述しています。
2. **az** CLI を使用してデプロイメントを作成します。

```
$ az deployment group create -g ${RESOURCE_GROUP} \  
--template-file "<installation_directory>/01_vnet.json" \  
--parameters baseName="${INFRA_ID}" 1
```

- 1** リソース名で使われるベース名。これは通常クラスターのインフラストラクチャー ID です。

3. VNet テンプレートをプライベート DNS ゾーンにリンクします。

```
$ az network private-dns link vnet create -g ${RESOURCE_GROUP} -z  
${CLUSTER_NAME}.${BASE_DOMAIN} -n ${INFRA_ID}-network-link -v "${INFRA_ID}-vnet"  
-e false
```

1.7.10.1. VNet の ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用し、OpenShift Container Platform クラスターに必要な VNet をデプロイすることができます。

例1.101_vnet.json ARM テンプレート

```

{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "addressPrefix" : "10.0.0.0/16",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetPrefix" : "10.0.0.0/24",
    "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
    "nodeSubnetPrefix" : "10.0.1.0/24",
    "clusterNsgName" : "[concat(parameters('baseName'), '-nsg')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/virtualNetworks",
      "name" : "[variables('virtualNetworkName')]",
      "location" : "[variables('location')]",
      "dependsOn" : [
        "[concat('Microsoft.Network/networkSecurityGroups/', variables('clusterNsgName'))]"
      ],
      "properties" : {
        "addressSpace" : {
          "addressPrefixes" : [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets" : [
          {
            "name" : "[variables('masterSubnetName')]",
            "properties" : {
              "addressPrefix" : "[variables('masterSubnetPrefix')]",
              "serviceEndpoints": [],
              "networkSecurityGroup" : {
                "id" : "[resourceId('Microsoft.Network/networkSecurityGroups', variables('clusterNsgName'))]"
              }
            }
          },
          {
            "name" : "[variables('nodeSubnetName')]",
            "properties" : {
              "addressPrefix" : "[variables('nodeSubnetPrefix')]",

```

```

        "serviceEndpoints": [],
        "networkSecurityGroup": {
            "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('clusterNsgName'))]"
        }
    }
}
]
}
},
{
    "type": "Microsoft.Network/networkSecurityGroups",
    "name": "[variables('clusterNsgName')]",
    "apiVersion": "2018-10-01",
    "location": "[variables('location')]",
    "properties": {
        "securityRules": [
            {
                "name": "apiserver_in",
                "properties": {
                    "protocol": "Tcp",
                    "sourcePortRange": "*",
                    "destinationPortRange": "6443",
                    "sourceAddressPrefix": "*",
                    "destinationAddressPrefix": "*",
                    "access": "Allow",
                    "priority": 101,
                    "direction": "Inbound"
                }
            }
        ]
    }
}
]
}
]
}

```

1.7.11. Azure インフラストラクチャー用の RHCOS クラスタイメージのデプロイ

OpenShift Container Platform ノードに Microsoft Azure 用の有効な Red Hat Enterprise Linux CoreOS (RHCOS) イメージを使用する必要があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- RHCOS 仮想ハードディスク (VHD) クラスタイメージを Azure ストレージコンテナに保存します。
- ブートストラップ Ignition 設定ファイルを Azure ストレージコンテナに保存します。

手順

1. 本トピックの **イメージストレージの ARM テンプレート** セクションからテンプレートをコピーし、これを **02_storage.json** としてクラスターのインストールディレクトリーに保存します。このテンプレートは、クラスターに必要なイメージストレージについて記述しています。
2. RHCOS VHD blob URL を変数としてエクスポートします。

```
$ export VHD_BLOB_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} -c vhd -n "rhcos.vhd" -o tsv`
```

3. クラスターイメージのデプロイ

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/02_storage.json" \
  --parameters vhdBlobURL="${VHD_BLOB_URL}" \ 1
  --parameters baseName="${INFRA_ID}" 2
```

- 1** マスターマシンおよびワーカーマシンを作成するために使用される RHCOS VHD の blob URL。
- 2** リソース名で使われるベース名。これは通常クラスターのインフラストラクチャー ID です。

1.7.11.1. イメージストレージの ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用し、OpenShift Container Platform クラスターに必要な保存された Red Hat Enterprise Linux CoreOS (RHCOS) をデプロイすることができます。

例1.2 02_storage.json ARM テンプレート

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "vhdBlobURL" : {
      "type" : "string",
      "metadata" : {
        "description" : "URL pointing to the blob where the VHD to be used to create master and worker machines is located"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "imageName" : "[concat(parameters('baseName'), '-image')]"
  },
  "resources" : [
```

```

{
  "apiVersion": "2018-06-01",
  "type": "Microsoft.Compute/images",
  "name": "[variables('imageName')]",
  "location": "[variables('location')]",
  "properties": {
    "storageProfile": {
      "osDisk": {
        "osType": "Linux",
        "osState": "Generalized",
        "blobUri": "[parameters('vhdBlobURL')]",
        "storageAccountType": "Standard_LRS"
      }
    }
  }
}
]
}

```

1.7.12. ユーザーによってプロビジョニングされるインフラストラクチャーのネットワーク要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

マシン間のネットワーク接続を、クラスターのコンポーネントが通信できるように設定する必要があります。すべてのマシンではクラスターの他のすべてのマシンのホスト名を解決できる必要があります。

表1.20 すべてのマシンに対応するすべてのマシン

プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	1936	メトリクス
	9000-9999	ホストレベルのサービス。ポート 9100-9101 のノードエクスポーター、ポート 9099 の Cluster Version Operator が含まれます。
	10250-10259	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	4789	VXLAN および Geneve
	6081	VXLAN および Geneve
	9000-9999	ポート 9100-9101 のノードエクスポーターを含む、ホストレベルのサービス。

プロトコル	ポート	説明
TCP/UDP	30000-32767	Kubernetes ノードポート

表1.21 コントロールプレーンへのすべてのマシン

プロトコル	ポート	説明
TCP	6443	Kubernetes API

表1.22 コントロールプレーンマシンへのコントロールプレーンマシン

プロトコル	ポート	説明
TCP	2379-2380	etcd サーバーおよびピアポート

ネットワークポロジータン要件

クラスター用にプロビジョニングするインフラストラクチャーは、ネットワークポロジータンの以下の要件を満たす必要があります。



重要

OpenShift Container Platform では、すべてのノードが、プラットフォームコンテナのイメージをプルし、Telemetry データを Red Hat に提供するためにインターネットへの直接のアクセスが必要です。

ロードバランサー

OpenShift Container Platform をインストールする前に、以下の要件を満たす 2 つのロードバランサーをプロビジョニングする必要があります。

1. **API ロードバランサー:** プラットフォームと対話およびプラットフォームを設定するためのユーザー向けの共通のエンドポイントを提供します。以下の条件を設定します。
 - Layer 4 の負荷分散のみ。これは、Raw TCP、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、API ルートの Server Name Indication (SNI) を有効にする必要があります。
 - ステートレス負荷分散アルゴリズム。オプションは、ロードバランサーの実装によって異なります。



注記

API ロードバランサーが適切に機能するには、セッション永続性は必要ありません。

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.23 API ロードバランサー

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明
6443	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。API サーバーのヘルスチェックプローブの /readyz エンドポイントを設定する必要があります。	X	X	Kubernetes API サーバー
22623	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。	X		マシン設定サーバー



注記

ロードバランサーは、API サーバーが **/readyz** エンドポイントをオフにしてからプールから API サーバーインスタンスを削除するまで最大 30 秒かかるように設定する必要があります。**/readyz** の後の時間枠内でエラーが返されたり、正常になったりする場合は、エンドポイントが削除または追加されているはずです。5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これらは十分にテストされた値です。

2. Application Ingress ロードバランサー: クラスター外から送られるアプリケーショントラフィックの Ingress ポイントを提供します。以下の条件を設定します。

- Layer 4 の負荷分散のみ。これは、Raw TCP、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、Ingress ルートの Server Name Indication (SNI) を有効にする必要があります。
- 選択可能なオプションやプラットフォーム上でホストされるアプリケーションの種類に基づいて、接続ベースの永続化またはセッションベースの永続化が推奨されます。

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.24 アプリケーション Ingress ロードバランサー

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明
443	デフォルトで Ingress ルーター Pod、コンピュート、またはワーカーを実行するマシン。	X	X	HTTPS トラフィック

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明
80	デフォルトで Ingress ルーター Pod、コンピュート、またはワーカーを実行するマシン。	X	X	HTTP トラフィック

ヒント

クライアントの実際の IP アドレスがロードバランサーによって確認できる場合、ソースの IP ベースのセッション永続化を有効にすると、エンドツーエンドの TLS 暗号化を使用するアプリケーションのパフォーマンスを強化できます。



注記

Ingress ルーターの作業用の設定が OpenShift Container Platform クラスターに必要です。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

1.7.13. Azure でのネットワークおよび負荷分散コンポーネントの作成

OpenShift Container Platform クラスターで使用するネットワークおよび負荷分散を Microsoft Azure で設定する必要があります。これらのコンポーネントを作成する方法として、提供される Azure Resource Manager (ARM) テンプレートを変更することができます。



注記

提供される ARM テンプレートを使用して Azure インフラストラクチャーを使用しない場合、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせる必要がある可能性があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- Azure で VNet および関連するサブネットを作成し、設定します。

手順

1. 本トピックの **ネットワークおよびロードバランサーの ARM テンプレート** セクションからテンプレートをコピーし、これを **03_infra.json** としてクラスターのインストールディレクトリーに保存します。このテンプレートは、クラスターに必要なネットワークおよび負荷分散オブジェクトについて記述しています。
2. **az** CLI を使用してデプロイメントを作成します。

```
$ az deployment group create -g ${RESOURCE_GROUP} \
--template-file "<installation_directory>/03_infra.json" \
--parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" ❶
--parameters baseName="${INFRA_ID}" ❷
```

- 1 プライベート DNS ゾーンの名前。
 - 2 リソース名で使われるベース名。これは通常クラスターのインフラストラクチャー ID です。
3. API パブリックロードバランサーのパブリックゾーンに **api** DNS レコードを作成します。 `${BASE_DOMAIN_RESOURCE_GROUP}` 変数は、パブリック DNS ゾーンがあるリソースグループをポイントする必要があります。

- a. 以下の変数をエクスポートします。

```
$ export PUBLIC_IP=`az network public-ip list -g ${RESOURCE_GROUP} --query "[?name=='${INFRA_ID}-master-pip'] | [0].ipAddress" -o tsv`
```

- b. 新しいパブリックゾーンに DNS レコードを作成します。

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n api -a ${PUBLIC_IP} --ttl 60
```

- c. クラスターを既存のパブリックゾーンに追加する場合は、DNS レコードを代わりに作成できます。

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${BASE_DOMAIN} -n api.${CLUSTER_NAME} -a ${PUBLIC_IP} --ttl 60
```

1.7.13.1. ネットワークおよびロードバランサーの ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用して、OpenShift Container Platform クラスターに必要なネットワークオブジェクトおよびロードバランサーをデプロイすることができます。

例1.3 03_infra.json ARM テンプレート

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "baseName": {
      "type": "string",
      "minLength": 1,
      "metadata": {
        "description": "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "privateDNSZoneName": {
      "type": "string",
      "metadata": {
        "description": "Name of the private DNS zone"
      }
    }
  },
  "variables": {
    "location": "[resourceGroup().location]",
```

```

    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterPublicIpAddressName" : "[concat(parameters('baseName'), '-master-pip')]",
    "masterPublicIpAddressID" : "[resourceId('Microsoft.Network/publicIPAddresses',
variables('masterPublicIpAddressName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "masterLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers',
variables('masterLoadBalancerName'))]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "internalLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers',
variables('internalLoadBalancerName'))]",
    "skuName": "Standard"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/publicIPAddresses",
      "name" : "[variables('masterPublicIpAddressName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "properties" : {
        "publicIPAllocationMethod" : "Static",
        "dnsSettings" : {
          "domainNameLabel" : "[variables('masterPublicIpAddressName')]"
        }
      }
    },
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/loadBalancers",
      "name" : "[variables('masterLoadBalancerName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "dependsOn" : [
        "[concat('Microsoft.Network/publicIPAddresses/', variables('masterPublicIpAddressName'))]"
      ],
      "properties" : {
        "frontendIPConfigurations" : [
          {
            "name" : "public-lb-ip",
            "properties" : {
              "publicIPAddress" : {
                "id" : "[variables('masterPublicIpAddressID')]"
              }
            }
          }
        ]
      }
    },
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/loadBalancers",
      "name" : "[variables('internalLoadBalancerName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "dependsOn" : [
        "[concat('Microsoft.Network/publicIPAddresses/', variables('masterPublicIpAddressName'))]"
      ],
      "properties" : {
        "frontendIPConfigurations" : [
          {
            "name" : "internal-lb-ip",
            "properties" : {
              "publicIPAddress" : {
                "id" : "[variables('masterPublicIpAddressID')]"
              }
            }
          }
        ]
      }
    }
  ],
  "backendAddressPools" : [

```

```

    {
      "name" : "public-lb-backend"
    }
  ],
  "loadBalancingRules" : [
    {
      "name" : "api-internal",
      "properties" : {
        "frontendIPConfiguration" : {
          "id" : "[concat(variables('masterLoadBalancerID'), '/frontendIPConfigurations/public-lb-
ip')]"
        },
        "backendAddressPool" : {
          "id" : "[concat(variables('masterLoadBalancerID'), '/backendAddressPools/public-lb-
backend')]"
        },
        "protocol" : "Tcp",
        "loadDistribution" : "Default",
        "idleTimeoutInMinutes" : 30,
        "frontendPort" : 6443,
        "backendPort" : 6443,
        "probe" : {
          "id" : "[concat(variables('masterLoadBalancerID'), '/probes/api-internal-probe')]"
        }
      }
    }
  ],
  "probes" : [
    {
      "name" : "api-internal-probe",
      "properties" : {
        "protocol" : "Https",
        "port" : 6443,
        "requestPath" : "/readyz",
        "intervalInSeconds" : 10,
        "numberOfProbes" : 3
      }
    }
  ]
},
{
  "apiVersion" : "2018-12-01",
  "type" : "Microsoft.Network/loadBalancers",
  "name" : "[variables('internalLoadBalancerName')]",
  "location" : "[variables('location')]",
  "sku" : {
    "name" : "[variables('skuName')]"
  },
  "properties" : {
    "frontendIPConfigurations" : [
      {
        "name" : "internal-lb-ip",
        "properties" : {
          "privateIPAllocationMethod" : "Dynamic",
          "subnet" : {

```

```

        "id" : "[variables('masterSubnetRef')]"
      },
      "privateIPAddressVersion" : "IPv4"
    }
  ],
  "backendAddressPools" : [
    {
      "name" : "internal-lb-backend"
    }
  ],
  "loadBalancingRules" : [
    {
      "name" : "api-internal",
      "properties" : {
        "frontendIPConfiguration" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
        },
        "frontendPort" : 6443,
        "backendPort" : 6443,
        "enableFloatingIP" : false,
        "idleTimeoutInMinutes" : 30,
        "protocol" : "Tcp",
        "enableTcpReset" : false,
        "loadDistribution" : "Default",
        "backendAddressPool" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
        },
        "probe" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/probes/api-internal-probe')]"
        }
      }
    },
    {
      "name" : "sint",
      "properties" : {
        "frontendIPConfiguration" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
        },
        "frontendPort" : 22623,
        "backendPort" : 22623,
        "enableFloatingIP" : false,
        "idleTimeoutInMinutes" : 30,
        "protocol" : "Tcp",
        "enableTcpReset" : false,
        "loadDistribution" : "Default",
        "backendAddressPool" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
        },
        "probe" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/probes/sint-probe')]"
        }
      }
    }
  ]
}

```

```

    }
  },
  ],
  "probes" : [
    {
      "name" : "api-internal-probe",
      "properties" : {
        "protocol" : "Https",
        "port" : 6443,
        "requestPath" : "/readyz",
        "intervalInSeconds" : 10,
        "numberOfProbes" : 3
      }
    },
    {
      "name" : "sint-probe",
      "properties" : {
        "protocol" : "Https",
        "port" : 22623,
        "requestPath" : "/healthz",
        "intervalInSeconds" : 10,
        "numberOfProbes" : 3
      }
    }
  ]
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api')]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
      }
    ]
  }
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api-int')]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
  "properties": {
    "ttl": 60,

```

```

    "aRecords": [
      {
        "ipv4Address": "[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
      }
    ]
  }
}
]
}
}

```

1.7.14. Azure でのブートストラップマシンの作成

OpenShift Container Platform クラスターの初期化を実行する際に使用するブートストラップマシンを Microsoft Azure で作成する必要があります。このマシンを作成する方法として、提供される Azure Resource Manager (ARM) テンプレートを変更することができます。



注記

提供されている ARM テンプレートを使用してブートストラップマシンを作成しない場合、指定される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせる必要がある可能性があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- Azure で VNet および関連するサブネットを作成し、設定します。
- Azure でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートリールを作成します。

手順

1. 本トピックの **ブートストラップマシンの ARM テンプレート** セクションからテンプレートをコピーし、これを **04_bootstrap.json** としてクラスターのインストールディレクトリーに保存します。このテンプレートは、クラスターに必要なブートストラップマシンについて記述しています。
2. ブートストラップマシンのデプロイメントに必要な以下の変数をエクスポートします。

```

$ export BOOTSTRAP_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} -c "files" -n "bootstrap.ign" -o tsv`
$ export BOOTSTRAP_IGNITION=`jq -rcnM --arg v "2.2.0" --arg url ${BOOTSTRAP_URL}
'{ignition:{version:$v,config:{replace:{source:$url}}}}' | base64 -w0`

```

3. **az** CLI を使用してデプロイメントを作成します。

```
$ az deployment group create -g ${RESOURCE_GROUP} \
--template-file "<installation_directory>/04_bootstrap.json" \
--parameters bootstrapIgnition="${BOOTSTRAP_IGNITION}" \ ❶
--parameters sshKeyData="${SSH_KEY}" \ ❷
--parameters baseName="${INFRA_ID}" ❸
```

- ❶ ブートストラップクラスターのブートストラップ Ignition コンテンツ。
- ❷ 文字列としての SSH RSA 公開鍵ファイル。
- ❸ リソース名で使われるベース名。これは通常クラスターのインフラストラクチャー ID です。

1.7.14.1. ブートストラップマシンの ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用し、OpenShift Container Platform クラスターに必要なブートストラップマシンをデプロイすることができます。

例1.4 04_bootstrap.json ARM テンプレート

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "bootstrapIgnition" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Bootstrap ignition content for the bootstrap cluster"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string."
      }
    },
    "bootstrapVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D4s_v3",
      "allowedValues" : [
        "Standard_A2",
        "Standard_A3",
        "Standard_A4",
        "Standard_A5",
        "Standard_A6",
```

```
"Standard_A7",
"Standard_A8",
"Standard_A9",
"Standard_A10",
"Standard_A11",
"Standard_D2",
"Standard_D3",
"Standard_D4",
"Standard_D11",
"Standard_D12",
"Standard_D13",
"Standard_D14",
"Standard_D2_v2",
"Standard_D3_v2",
"Standard_D4_v2",
"Standard_D5_v2",
"Standard_D8_v3",
"Standard_D11_v2",
"Standard_D12_v2",
"Standard_D13_v2",
"Standard_D14_v2",
"Standard_E2_v3",
"Standard_E4_v3",
"Standard_E8_v3",
"Standard_E16_v3",
"Standard_E32_v3",
"Standard_E64_v3",
"Standard_E2s_v3",
"Standard_E4s_v3",
"Standard_E8s_v3",
"Standard_E16s_v3",
"Standard_E32s_v3",
"Standard_E64s_v3",
"Standard_G1",
"Standard_G2",
"Standard_G3",
"Standard_G4",
"Standard_G5",
"Standard_DS2",
"Standard_DS3",
"Standard_DS4",
"Standard_DS11",
"Standard_DS12",
"Standard_DS13",
"Standard_DS14",
"Standard_DS2_v2",
"Standard_DS3_v2",
"Standard_DS4_v2",
"Standard_DS5_v2",
"Standard_DS11_v2",
"Standard_DS12_v2",
"Standard_DS13_v2",
"Standard_DS14_v2",
"Standard_GS1",
"Standard_GS2",
"Standard_GS3",
```

```

    "Standard_GS4",
    "Standard_GS5",
    "Standard_D2s_v3",
    "Standard_D4s_v3",
    "Standard_D8s_v3"
  ],
  "metadata" : {
    "description" : "The size of the Bootstrap Virtual Machine"
  }
},
"variables" : {
  "location" : "[resourceGroup().location]",
  "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
  "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
  "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
  "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
  "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
  "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
  "sshKeyPath" : "/home/core/.ssh/authorized_keys",
  "identityName" : "[concat(parameters('baseName'), '-identity')]",
  "vmName" : "[concat(parameters('baseName'), '-bootstrap')]",
  "nicName" : "[concat(variables('vmName'), '-nic')]",
  "imageName" : "[concat(parameters('baseName'), '-image')]",
  "clusterNsgName" : "[concat(parameters('baseName'), '-nsg')]",
  "sshPublicIpAddressName" : "[concat(variables('vmName'), '-ssh-pip')]"
},
"resources" : [
  {
    "apiVersion" : "2018-12-01",
    "type" : "Microsoft.Network/publicIPAddresses",
    "name" : "[variables('sshPublicIpAddressName')]",
    "location" : "[variables('location')]",
    "sku" : {
      "name": "Standard"
    },
    "properties" : {
      "publicIPAllocationMethod" : "Static",
      "dnsSettings" : {
        "domainNameLabel" : "[variables('sshPublicIpAddressName')]"
      }
    }
  },
  {
    "apiVersion" : "2018-06-01",
    "type" : "Microsoft.Network/networkInterfaces",
    "name" : "[variables('nicName')]",
    "location" : "[variables('location')]",
    "dependsOn" : [
      "[resourceId('Microsoft.Network/publicIPAddresses', variables('sshPublicIpAddressName'))]"
    ],
    "properties" : {
      "ipConfigurations" : [
        {

```

```

    "name" : "pipConfig",
    "properties" : {
      "privateIPAllocationMethod" : "Dynamic",
      "publicIPAddress": {
        "id": "[resourceId('Microsoft.Network/publicIPAddresses',
variables('sshPublicIpAddressName'))]"
      },
      "subnet" : {
        "id" : "[variables('masterSubnetRef')]"
      },
      "loadBalancerBackendAddressPools" : [
        {
          "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
        },
        {
          "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
        }
      ]
    }
  }
}
],
{
  "apiVersion" : "2018-06-01",
  "type" : "Microsoft.Compute/virtualMachines",
  "name" : "[variables('vmName')]",
  "location" : "[variables('location')]",
  "identity" : {
    "type" : "userAssigned",
    "userAssignedIdentities" : {
      "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
    }
  },
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
  ],
  "properties" : {
    "hardwareProfile" : {
      "vmSize" : "[parameters('bootstrapVMSize')]"
    },
    "osProfile" : {
      "computerName" : "[variables('vmName')]",
      "adminUsername" : "core",
      "customData" : "[parameters('bootstrapIgnition')]",
      "linuxConfiguration" : {
        "disablePasswordAuthentication" : true,
        "ssh" : {
          "publicKeys" : [
            {
              "path" : "[variables('sshKeyPath')]",

```

```

        "keyData" : "[parameters('sshKeyData')]"
      }
    ]
  }
},
"storageProfile" : {
  "imageReference": {
    "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
  },
  "osDisk" : {
    "name": "[concat(variables('vmName'), '_OSDisk')]",
    "osType" : "Linux",
    "createOption" : "FromImage",
    "managedDisk": {
      "storageAccountType": "Premium_LRS"
    },
    "diskSizeGB" : 100
  }
},
"networkProfile" : {
  "networkInterfaces" : [
    {
      "id" : "[resourceId('Microsoft.Network/networkInterfaces', variables('nicName'))]"
    }
  ]
}
},
{
  "apiVersion" : "2018-06-01",
  "type": "Microsoft.Network/networkSecurityGroups/securityRules",
  "name" : "[concat(variables('clusterNsgName'), '/bootstrap_ssh_in')]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[resourceId('Microsoft.Compute/virtualMachines', variables('vmName'))]"
  ],
  "properties": {
    "protocol" : "Tcp",
    "sourcePortRange" : "*",
    "destinationPortRange" : "22",
    "sourceAddressPrefix" : "*",
    "destinationAddressPrefix" : "*",
    "access" : "Allow",
    "priority" : 100,
    "direction" : "Inbound"
  }
}
]
}

```

1.7.15. Azure でのコントロールプレーンの作成

クラスターで使用するコントロールプレーンマシンを Microsoft Azure で作成する必要があります。これらのマシンを作成する方法として、提供される Azure Resource Manager (ARM) テンプレートを変更することができます。



注記

提供される ARM テンプレートを使用してコントロールプレーンマシンを使用しない場合、指定される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせる必要がある可能性があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- Azure で VNet および関連するサブネットを作成し、設定します。
- Azure でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートリールを作成します。
- ブートストラップマシンを作成します。

手順

1. 本トピックの **コントロールプレーンマシンの ARM テンプレート** セクションからテンプレートをコピーし、これを **05_masters.json** としてクラスターのインストールディレクトリに保存します。このテンプレートは、クラスターに必要なコントロールプレーンのマシンについて記述しています。
2. コントロールプレーンマシンのデプロイメントに必要な以下の変数をエクスポートします。

```
$ export MASTER_IGNITION=`cat <installation_directory>/master.ign | base64`
```

3. **az** CLI を使用してデプロイメントを作成します。

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/05_masters.json" \
  --parameters masterIgnition="${MASTER_IGNITION}" ① \
  --parameters sshKeyData="${SSH_KEY}" ② \
  --parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" ③ \
  --parameters baseName="${INFRA_ID}" ④
```

- ① マスターノードの Ignition コンテンツ。
- ② 文字列としての SSH RSA 公開鍵ファイル。
- ③ マスターノードが割り当てられているプライベート DNS ゾーンの名前。
- ④ リソース名で使用するベース名。これは通常クラスターのインフラストラクチャー ID です。

1.7.15.1. コントロールプレーンマシンの ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用し、OpenShift Container Platform クラスターに必要なコントロールプレーンマシンをデプロイすることができます。

例1.5 05_masters.json ARM テンプレート

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "masterIgnition" : {
      "type" : "string",
      "metadata" : {
        "description" : "Ignition content for the master nodes"
      }
    },
    "numberOfMasters" : {
      "type" : "int",
      "defaultValue" : 3,
      "minValue" : 2,
      "maxValue" : 30,
      "metadata" : {
        "description" : "Number of OpenShift masters to deploy"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string"
      }
    },
    "privateDNSZoneName" : {
      "type" : "string",
      "metadata" : {
        "description" : "Name of the private DNS zone the master nodes are going to be attached to"
      }
    },
    "masterVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D8s_v3",
      "allowedValues" : [
        "Standard_A2",
        "Standard_A3",
        "Standard_A4",
        "Standard_A5",
        "Standard_A6",
        "Standard_A7",
```

```
"Standard_A8",
"Standard_A9",
"Standard_A10",
"Standard_A11",
"Standard_D2",
"Standard_D3",
"Standard_D4",
"Standard_D11",
"Standard_D12",
"Standard_D13",
"Standard_D14",
"Standard_D2_v2",
"Standard_D3_v2",
"Standard_D4_v2",
"Standard_D5_v2",
"Standard_D8_v3",
"Standard_D11_v2",
"Standard_D12_v2",
"Standard_D13_v2",
"Standard_D14_v2",
"Standard_E2_v3",
"Standard_E4_v3",
"Standard_E8_v3",
"Standard_E16_v3",
"Standard_E32_v3",
"Standard_E64_v3",
"Standard_E2s_v3",
"Standard_E4s_v3",
"Standard_E8s_v3",
"Standard_E16s_v3",
"Standard_E32s_v3",
"Standard_E64s_v3",
"Standard_G1",
"Standard_G2",
"Standard_G3",
"Standard_G4",
"Standard_G5",
"Standard_DS2",
"Standard_DS3",
"Standard_DS4",
"Standard_DS11",
"Standard_DS12",
"Standard_DS13",
"Standard_DS14",
"Standard_DS2_v2",
"Standard_DS3_v2",
"Standard_DS4_v2",
"Standard_DS5_v2",
"Standard_DS11_v2",
"Standard_DS12_v2",
"Standard_DS13_v2",
"Standard_DS14_v2",
"Standard_GS1",
"Standard_GS2",
"Standard_GS3",
"Standard_GS4",
```

```

    "Standard_GS5",
    "Standard_D2s_v3",
    "Standard_D4s_v3",
    "Standard_D8s_v3"
  ],
  "metadata" : {
    "description" : "The size of the Master Virtual Machines"
  }
},
"diskSizeGB" : {
  "type" : "int",
  "defaultValue" : 1024,
  "metadata" : {
    "description" : "Size of the Master VM OS disk, in GB"
  }
}
},
"variables" : {
  "location" : "[resourceGroup().location]",
  "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
  "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
  "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
  "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
  "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
  "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
  "sshKeyPath" : "/home/core/.ssh/authorized_keys",
  "identityName" : "[concat(parameters('baseName'), '-identity')]",
  "imageName" : "[concat(parameters('baseName'), '-image')]",
  "copy" : [
    {
      "name" : "vmNames",
      "count" : "[parameters('numberOfMasters')]",
      "input" : "[concat(parameters('baseName'), '-master-', copyIndex('vmNames'))]"
    }
  ]
},
"resources" : [
  {
    "apiVersion" : "2018-06-01",
    "type" : "Microsoft.Network/networkInterfaces",
    "copy" : {
      "name" : "nicCopy",
      "count" : "[length(variables('vmNames'))]"
    },
    "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",
    "location" : "[variables('location')]",
    "properties" : {
      "ipConfigurations" : [
        {
          "name" : "pipConfig",
          "properties" : {
            "privateIPAllocationMethod" : "Dynamic",
            "subnet" : {
              "id" : "[variables('masterSubnetRef')]"
            }
          }
        }
      ]
    }
  }
]
}

```

```

    },
    "loadBalancerBackendAddressPools" : [
      {
        "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
      },
      {
        "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
      }
    ]
  }
}
]
}
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/SRV",
  "name": "[concat(parameters('privateDNSZoneName'), '/_etcd-server-ssl._tcp')]",
  "location" : "[variables('location')]",
  "properties": {
    "ttl": 60,
    "copy": [{
      "name": "srvRecords",
      "count": "[length(variables('vmNames'))]",
      "input": {
        "priority": 0,
        "weight" : 10,
        "port" : 2380,
        "target" : "[concat('etcd-', copyIndex('srvRecords'), '.',
parameters('privateDNSZoneName'))]"
      }
    }]
  }
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "copy" : {
    "name" : "dnsCopy",
    "count" : "[length(variables('vmNames'))]"
  },
  "name": "[concat(parameters('privateDNSZoneName'), '/etcd-', copyIndex())]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-
nic'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "[reference(concat(variables('vmNames')[copyIndex()], '-

```

```

nic')).ipConfigurations[0].properties.privateIpAddress]"
    }
  ]
}
},
{
  "apiVersion" : "2018-06-01",
  "type" : "Microsoft.Compute/virtualMachines",
  "copy" : {
    "name" : "vmCopy",
    "count" : "[length(variables('vmNames'))]"
  },
  "name" : "[variables('vmNames')[copyIndex()]]",
  "location" : "[variables('location')]",
  "identity" : {
    "type" : "userAssigned",
    "userAssignedIdentities" : {
      "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
    }
  },
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-nic'))]",
    "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'), '/A/etcd-', copyIndex()])]",
    "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'), '/SRV/_etcd-server-ssl._tcp')]"
  ],
  "properties" : {
    "hardwareProfile" : {
      "vmSize" : "[parameters('masterVMSize')]"
    },
    "osProfile" : {
      "computerName" : "[variables('vmNames')[copyIndex()]]",
      "adminUsername" : "core",
      "customData" : "[parameters('masterIgnition')]",
      "linuxConfiguration" : {
        "disablePasswordAuthentication" : true,
        "ssh" : {
          "publicKeys" : [
            {
              "path" : "[variables('sshKeyPath')]",
              "keyData" : "[parameters('sshKeyData')]"
            }
          ]
        }
      }
    },
    "storageProfile" : {
      "imageReference": {
        "id": "[resourceID('Microsoft.Compute/images', variables('imageName'))]"
      },
      "osDisk" : {
        "name": "[concat(variables('vmNames')[copyIndex()], '_OSDisk')]",
        "osType" : "Linux",

```

```

        "createOption" : "FromImage",
        "caching": "ReadOnly",
        "writeAcceleratorEnabled": false,
        "managedDisk": {
            "storageAccountType": "Premium_LRS"
        },
        "diskSizeGB" : "[parameters('diskSizeGB')]"
    },
    },
    "networkProfile" : {
        "networkInterfaces" : [
            {
                "id" : "[resourceId('Microsoft.Network/networkInterfaces', concat(variables('vmNames')
[copyIndex()], '-nic'))]",
                "properties": {
                    "primary": false
                }
            }
        ]
    }
}
]
}
}

```

1.7.16. ブートストラップの完了を待機し、Azure のブートストラップリソースを削除する

Microsoft Azure ですべての必要なインフラストラクチャーを作成した後に、ブートストラッププロセスが、インストールプログラムで生成した Ignition 設定ファイルを使用してプロビジョニングしたマシンで完了するのを待機します。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- Azure で VNet および関連するサブネットを作成し、設定します。
- Azure でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートリールを作成します。
- ブートストラップマシンを作成します。
- コントロールプレーンマシンを作成します。

手順

1. インストールプログラムが含まれるディレクトリーに切り替え、以下のコマンドを実行します。

```
$ ./openshift-install wait-for bootstrap-complete --dir=<installation_directory> \ ❶
--log-level info ❷
```

- ❶ <installation_directory> には、インストールファイルを保存したディレクトリへのパスを指定します。
- ❷ 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。

コマンドが **FATAL** 警告を出さずに終了する場合、実稼働用のコントロールプレーンは初期化されています。

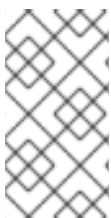
2. ブートストラップリソースを削除します。

```
$ az network nsg rule delete -g ${RESOURCE_GROUP} --nsg-name ${INFRA_ID}-nsg --
name bootstrap_ssh_in
$ az vm stop -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm deallocate -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap --yes
$ az disk delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap_OSDisk --no-
wait --yes
$ az network nic delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-nic --no-
wait
$ az storage blob delete --account-key ${ACCOUNT_KEY} --account-name
${CLUSTER_NAME}sa --container-name files --name bootstrap.ign
$ az network public-ip delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-
ssh-pip
```

1.7.17. Azure での追加のワーカーマシンの作成

Microsoft Azure でクラスターが使用するワーカーマシンを作成するには、それぞれのインスタンスを個別に起動するか、または自動スケーリンググループなどのクラスター外にある自動プロセスを実行します。OpenShift Container Platform の組み込まれたクラスタースケーリングメカニズムやマシン API を利用できます。

この例では、Azure Resource Manager (ARM) テンプレートを使用して1つのインスタンスを手動で起動します。追加のインスタンスは、ファイル内に **06_workers.json** というタイプのリソースを追加して起動することができます。



注記

提供される ARM テンプレートを使用してワーカーマシンを使用しない場合、指定される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせる必要がある可能性があります。

前提条件

- Azure アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- Azure で VNet および関連するサブネットを作成し、設定します。

- Azure でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートリールを作成します。
- ブートストラップマシンを作成します。
- コントロールプレーンマシンを作成します。

手順

1. 本トピックの **ワーカーマシンの ARM テンプレート** セクションからテンプレートをコピーし、これを **06_workers.json** としてクラスターのインストールディレクトリーに保存します。このテンプレートは、クラスターに必要なワーカーマシンについて記述しています。
2. ワーカーマシンのデプロイメントに必要な以下の変数をエクスポートします。

```
$ export WORKER_IGNITION=`cat <installation_directory>/worker.ign | base64`
```

3. **az** CLI を使用してデプロイメントを作成します。

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/06_workers.json" \
  --parameters workerIgnition="${WORKER_IGNITION}" \ ❶
  --parameters sshKeyData="${SSH_KEY}" \ ❷
  --parameters baseName="${INFRA_ID}" ❸
```

- ❶ ワーカーノードの Ignition コンテンツ。
- ❷ 文字列としての SSH RSA 公開鍵ファイル。
- ❸ リソース名で使われるベース名。これは通常クラスターのインフラストラクチャー ID です。

1.7.17.1. ワーカーマシンの ARM テンプレート

以下の Azure Resource Manager (ARM) テンプレートを使用し、OpenShift Container Platform クラスターに必要なワーカーマシンをデプロイすることができます。

例1.6 06_workers.json ARM テンプレート

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "workerIgnition" : {
      "type" : "string",
```

```
"metadata" : {
  "description" : "Ignition content for the worker nodes"
},
},
"numberOfNodes" : {
  "type" : "int",
  "defaultValue" : 3,
  "minValue" : 2,
  "maxValue" : 30,
  "metadata" : {
    "description" : "Number of OpenShift compute nodes to deploy"
  }
},
"sshKeyData" : {
  "type" : "securestring",
  "metadata" : {
    "description" : "SSH RSA public key file as a string"
  }
},
"nodeVMSize" : {
  "type" : "string",
  "defaultValue" : "Standard_D4s_v3",
  "allowedValues" : [
    "Standard_A2",
    "Standard_A3",
    "Standard_A4",
    "Standard_A5",
    "Standard_A6",
    "Standard_A7",
    "Standard_A8",
    "Standard_A9",
    "Standard_A10",
    "Standard_A11",
    "Standard_D2",
    "Standard_D3",
    "Standard_D4",
    "Standard_D11",
    "Standard_D12",
    "Standard_D13",
    "Standard_D14",
    "Standard_D2_v2",
    "Standard_D3_v2",
    "Standard_D4_v2",
    "Standard_D5_v2",
    "Standard_D8_v3",
    "Standard_D11_v2",
    "Standard_D12_v2",
    "Standard_D13_v2",
    "Standard_D14_v2",
    "Standard_E2_v3",
    "Standard_E4_v3",
    "Standard_E8_v3",
    "Standard_E16_v3",
    "Standard_E32_v3",
    "Standard_E64_v3",
    "Standard_E2s_v3",
```

```

        "Standard_E4s_v3",
        "Standard_E8s_v3",
        "Standard_E16s_v3",
        "Standard_E32s_v3",
        "Standard_E64s_v3",
        "Standard_G1",
        "Standard_G2",
        "Standard_G3",
        "Standard_G4",
        "Standard_G5",
        "Standard_DS2",
        "Standard_DS3",
        "Standard_DS4",
        "Standard_DS11",
        "Standard_DS12",
        "Standard_DS13",
        "Standard_DS14",
        "Standard_DS2_v2",
        "Standard_DS3_v2",
        "Standard_DS4_v2",
        "Standard_DS5_v2",
        "Standard_DS11_v2",
        "Standard_DS12_v2",
        "Standard_DS13_v2",
        "Standard_DS14_v2",
        "Standard_GS1",
        "Standard_GS2",
        "Standard_GS3",
        "Standard_GS4",
        "Standard_GS5",
        "Standard_D2s_v3",
        "Standard_D4s_v3",
        "Standard_D8s_v3"
    ],
    "metadata" : {
        "description" : "The size of the each Node Virtual Machine"
    }
}
},
"variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
    "nodeSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('nodeSubnetName'))]",
    "infraLoadBalancerName" : "[parameters('baseName')]",
    "sshKeyPath" : "/home/capi/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "copy" : [
        {
            "name" : "vmNames",
            "count" : "[parameters('numberOfNodes')]",
            "input" : "[concat(parameters('baseName'), '-worker-', variables('location'), '-',

```

```

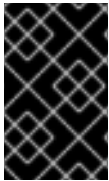
copyIndex('vmNames', 1))]"
    }
  ]
},
"resources" : [
  {
    "apiVersion" : "2019-05-01",
    "name" : "[concat('node', copyIndex())]",
    "type" : "Microsoft.Resources/deployments",
    "copy" : {
      "name" : "nodeCopy",
      "count" : "[length(variables('vmNames'))]"
    },
    "properties" : {
      "mode" : "Incremental",
      "template" : {
        "$schema" : "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
        "contentVersion" : "1.0.0.0",
        "resources" : [
          {
            "apiVersion" : "2018-06-01",
            "type" : "Microsoft.Network/networkInterfaces",
            "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",
            "location" : "[variables('location')]",
            "properties" : {
              "ipConfigurations" : [
                {
                  "name" : "pipConfig",
                  "properties" : {
                    "privateIPAllocationMethod" : "Dynamic",
                    "subnet" : {
                      "id" : "[variables('nodeSubnetRef')]"
                    }
                  }
                }
              ]
            }
          }
        ]
      }
    },
    "identity" : {
      "type" : "userAssigned",
      "userAssignedIdentities" : {
        "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
      }
    },
    "dependsOn" : [
      "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')

```

```
[copyIndex()], '-nic'))]"
],
"properties": {
  "hardwareProfile": {
    "vmSize": "[parameters('nodeVMSize')]"
  },
  "osProfile": {
    "computerName": "[variables('vmNames')[copyIndex()]]",
    "adminUsername": "capi",
    "customData": "[parameters('workerIgnition')]",
    "linuxConfiguration": {
      "disablePasswordAuthentication": true,
      "ssh": {
        "publicKeys": [
          {
            "path": "[variables('sshKeyPath')]",
            "keyData": "[parameters('sshKeyData')]"
          }
        ]
      }
    }
  },
  "storageProfile": {
    "imageReference": {
      "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
    },
    "osDisk": {
      "name": "[concat(variables('vmNames')[copyIndex()], '_OSDisk')]",
      "osType": "Linux",
      "createOption": "FromImage",
      "managedDisk": {
        "storageAccountType": "Premium_LRS"
      },
      "diskSizeGB": 128
    }
  },
  "networkProfile": {
    "networkInterfaces": [
      {
        "id": "[resourceId('Microsoft.Network/networkInterfaces',
concat(variables('vmNames')[copyIndex()], '-nic'))]",
        "properties": {
          "primary": true
        }
      }
    ]
  }
}
]
```

1.7.18. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。



重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.5 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.7.18.1. Linux への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.7.18.2. Windows での CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。

5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

1.7.18.3. macOS への CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.7.19. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイします。
- **oc** CLI をインストールします。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

1.7.20. マシンの証明書署名要求の承認

マシンをクラスターに追加する際に、追加したそれぞれのマシンについて2つの保留状態の証明書署名要求 (CSR) が生成されます。これらの CSR が承認されていることを確認するか、または必要な場合はそれらを承認してください。最初にクライアント要求を承認し、次にサーバー要求を承認する必要があります。

前提条件

- マシンがクラスターに追加されています。

手順

1. クラスターがマシンを認識していることを確認します。

```
$ oc get nodes
```

出力例

```
NAME      STATUS   ROLES    AGE   VERSION
master-0  Ready    master   63m   v1.18.3
master-1  Ready    master   63m   v1.18.3
master-2  Ready    master   64m   v1.18.3
worker-0  NotReady worker   76s   v1.18.3
worker-1  NotReady worker   70s   v1.18.3
```

出力には作成したすべてのマシンが一覧表示されます。

2. 保留中の証明書署名要求 (CSR) を確認し、クラスターに追加したそれぞれのマシンのクライアントおよびサーバー要求に **Pending** または **Approved** ステータスが表示されていることを確認します。

```
$ oc get csr
```

出力例

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-8b2br  15m   system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
```

```
csr-8vnps 15m system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
...
```

この例では、2つのマシンがクラスターに参加しています。この一覧にはさらに多くの承認された CSR が表示される可能性があります。

- 追加したマシンの保留中の CSR すべてが **Pending** ステータスになった後に CSR が承認されない場合には、クラスターマシンの CSR を承認します。



注記

CSR のローテーションは自動的に実行されるため、クラスターにマシンを追加後1時間以内に CSR を承認してください。1時間以内に承認しない場合には、証明書のローテーションが行われ、各ノードに3つ以上の証明書が存在するようになります。これらの証明書すべてを承認する必要があります。クライアントの CSR が承認されたら、Kubelet は提供証明書のセカンダリー CSR を作成します。これには、手動の承認が必要です。次に、後続の提供証明書の更新要求は、Kubelet が同じパラメーターを持つ新規証明書を要求する場合に **machine-approver** によって自動的に承認されます。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}
{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```

- クライアント要求が承認されたら、クラスターに追加した各マシンのサーバー要求を確認する必要があります。

```
$ oc get csr
```

出力例

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

- 残りの CSR が承認されず、それらが **Pending** ステータスにある場合、クラスターマシンの CSR を承認します。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

■

```
$ oc adm certificate approve <csr_name> ❶
```

❶ **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs oc adm certificate approve
```

- すべてのクライアントおよびサーバーの CSR が承認された後に、マシンのステータスが **Ready** になります。以下のコマンドを実行して、これを確認します。

```
$ oc get nodes
```

出力例

NAME	STATUS	ROLES	AGE	VERSION
master-0	Ready	master	73m	v1.20.0
master-1	Ready	master	73m	v1.20.0
master-2	Ready	master	74m	v1.20.0
worker-0	Ready	worker	11m	v1.20.0
worker-1	Ready	worker	11m	v1.20.0



注記

サーバー CSR の承認後にマシンが **Ready** ステータスに移行するまでに数分の時間がかかる場合があります。

関連情報

- CSR の詳細は、[Certificate Signing Requests](#) を参照してください。

1.7.21. Ingress DNS レコードの追加

Kubernetes マニフェストの作成および Ignition 設定の生成時に DNS ゾーン設定を削除した場合、Ingress ロードバランサーをポイントする DNS レコードを手動で作成する必要があります。ワイルドカード ***.apps.{baseDomain}**。または特定のレコードのいずれかを作成できます。要件に基づいて A、CNAME その他のレコードを使用できます。

前提条件

- 独自にプロビジョニングしたインフラストラクチャーを使用して、OpenShift Container Platform クラスターを Microsoft Azure にデプロイしています。
- OpenShift CLI (**oc**) をインストールします。
- jq** パッケージをインストールします。
- [Azure CLI](#) のインストールまたは更新を実行します。

手順

1. Ingress ルーターがロードバランサーを作成し、**EXTERNAL-IP** フィールドにデータを設定していることを確認します。

```
$ oc -n openshift-ingress get service router-default
```

出力例

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
router-default	LoadBalancer	172.30.20.10	35.130.120.110	80:32288/TCP,443:31215/TCP	20

2. Ingress ルーター IP を変数としてエクスポートします。

```
$ export PUBLIC_IP_ROUTER=`oc -n openshift-ingress get service router-default --no-headers | awk '{print $4}'`
```

3. パブリック DNS ゾーンに ***.apps** レコードを追加します。

- a. このクラスターを新しいパブリックゾーンに追加する場合は、以下を実行します。

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER} --ttl 300
```

- b. このクラスターを既存のパブリックゾーンに追加する場合は、以下を実行します。

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${BASE_DOMAIN} -n *.apps.${CLUSTER_NAME} -a ${PUBLIC_IP_ROUTER} --ttl 300
```

4. ***.apps** レコードをプライベート DNS ゾーンに追加します。

- a. 以下のコマンドを使用して ***.apps** レコードを作成します。

```
$ az network private-dns record-set a create -g ${RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps --ttl 300
```

- b. 以下のコマンドを使用して ***.apps** レコードをプライベート DNS ゾーンに追加します。

```
$ az network private-dns record-set a add-record -g ${RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER}
```

ワイルドカードを使用する代わりに明示的なドメインを追加する場合は、クラスターのそれぞれの現行ルートのエントリーを作成できます。

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}{"\n"}{end}{end}' routes
```

出力例

```
oauth-openshift.apps.cluster.basedomain.com
console-openshift-console.apps.cluster.basedomain.com
downloads-openshift-console.apps.cluster.basedomain.com
```

```

alertmanager-main-openshift-monitoring.apps.cluster.basedomain.com
grafana-openshift-monitoring.apps.cluster.basedomain.com
prometheus-k8s-openshift-monitoring.apps.cluster.basedomain.com

```

1.7.22. ユーザーによってプロビジョニングされるインフラストラクチャーでの Azure インストールの実行

Microsoft Azure のユーザーによってプロビジョニングされるインフラストラクチャーで OpenShift Container Platform のインストールを開始した後は、クラスターが準備状態になるまでクラスターのイベントをモニターできます。

前提条件

- OpenShift Container Platform クラスターのブートストラップマシンを、ユーザーによってプロビジョニングされる Azure インフラストラクチャーにデプロイします。
- **oc** CLI をインストールし、ログインします。

手順

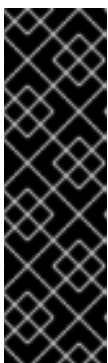
- クラスターのインストールを完了します。

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
```

出力例

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

- 1** **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** についてのドキュメントを参照してください。

1.8. AZURE でのクラスターのアンインストール

Microsoft Azure にデプロイしたクラスターは削除することができます。

1.8.1. インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターの削除

インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターは、クラウドから削除できます。



注記

アンインストール後に、とくにユーザーによってプロビジョニングされるインフラストラクチャー (UPI) クラスターで適切に削除されていないリソースがあるかどうかについて、クラウドプロバイダーを確認します。インストーラーが作成されなかったり、インストーラーがアクセスできない場合には、リソースがある可能性があります。

前提条件

- クラスターをデプロイするために使用したインストールプログラムのコピーがあります。
- クラスター作成時にインストールプログラムが生成したファイルがあります。

手順

1. クラスターをインストールするために使用したコンピューターから、以下のコマンドを実行します。

```
$ ./openshift-install destroy cluster \  
--dir=<installation_directory> --log-level=info ① ②
```

- ① **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。
- ② 異なる詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。



注記

クラスターのクラスター定義ファイルが含まれるディレクトリーを指定する必要があります。クラスターを削除するには、インストールプログラムでこのディレクトリーにある **metadata.json** ファイルが必要になります。

2. オプション: **<installation_directory>** ディレクトリーおよび OpenShift Container Platform インストールプログラムを削除します。