

Alibaba Cloud ApsaraDB for Redis

ユーザガイド

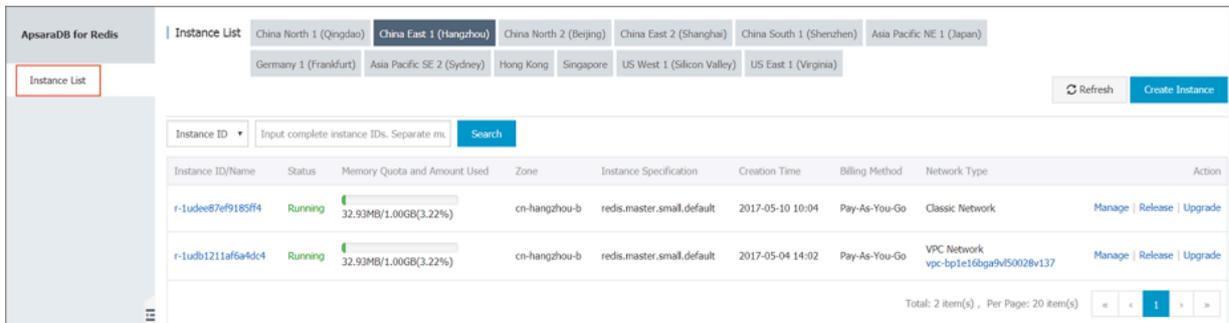
Document Version20190710

目次

1	コンソールへのログイン.....	1
2	インスタンス管理.....	2
2.1	パスワードの変更.....	2
2.2	IP ホワイトリストの設定.....	2
2.3	設定の変更.....	3
2.4	ネットワークタイプの設定.....	4
2.5	インスタンスの更新.....	5
2.6	メンテナンス時間の設定.....	6
2.7	マイナーバージョンのアップグレード.....	7
2.8	データの消去.....	8
2.9	インスタンスのリリース.....	8
3	接続管理.....	10
3.1	クラシックネットワークの保持期間の変更.....	10
4	バックアップとリカバリ.....	11
4.1	バックアップとリカバリ.....	11
5	データ移行.....	15
5.1	オンプレミス Redis から ApsaraDB Redis への移行.....	15
5.1.1	オンプレミス Redis クラスターから ApsaraDB Redis クラスターへの データ移行.....	15
5.1.2	redis-port を使用したデータの移行.....	17
5.1.3	Data Transmission Service (DTS) を使用したデータの移行.....	19
5.1.4	RDB ファイルを使用したオンプレミス Redis データの ApsaraDB Redis への移行.....	26
5.1.5	AOF を使用したオンプレミス Redis データの ApsaraDB Redis への移 行.....	27
5.2	ApsaraDB Redis からオンプレミス Redis への移行.....	28
5.2.1	バックアップセットの移行.....	28
5.3	ApsaraDB Redis インスタンス間の移行.....	29
5.3.1	redis-port ツールで異なるアカウントの ApsaraDB for Redis インスタ ンス間のデータの移行.....	30
5.3.2	Rump を使用した ApsaraDB Redis データベース間のデータ移行.....	31
6	アラーム設定.....	33

1 コンソールへのログイン

[こちら](#)をクリックして Redis コンソールにアクセスします。下図の示すように、インスタンスリストページが表示されます。



The screenshot displays the 'Instance List' page in the ApsaraDB for Redis console. The page features a navigation bar with various regions and a search bar. The main content area contains a table of instances. The 'Instance List' tab is highlighted in the left sidebar.

Instance ID/Name	Status	Memory Quota and Amount Used	Zone	Instance Specification	Creation Time	Billing Method	Network Type	Action
r-1udee87ef9185ff4	Running	32.93MB/1.00GB(3.22%)	cn-hangzhou-b	redis.master.small.default	2017-05-10 10:04	Pay-As-You-Go	Classic Network	Manage Release Upgrade
r-1udb1211af6a4dc4	Running	32.93MB/1.00GB(3.22%)	cn-hangzhou-b	redis.master.small.default	2017-05-04 14:02	Pay-As-You-Go	VPC Network vpc-bp1e16bga9v50028v137	Manage Release Upgrade

Total: 2 item(s) , Per Page: 20 item(s)

2 インスタンス管理

2.1 パスワードの変更

パスワードを忘れた場合や変更する必要がある場合、またはインスタンスの作成時にパスワードを設定しなかった場合は、インスタンスパスワードをリセットします。

1. [Redis コンソール](#)にログインします。
2. ターゲットインスタンスを見つけ、インスタンス ID または [管理] をクリックして [インスタンス情報] ページに移動します。
3. [パスワードの変更] ウィンドウで、古いパスワードと新しいパスワードを入力し、[OK] をクリックします。



:

- ・ パスワードを忘れた場合は、[パスワードの変更] ウィンドウで [パスワードを忘れた場合] をクリックします。次に [パスワードのリセット] ウィンドウで新しいパスワードを設定します。
- ・ パスワードは大文字、小文字、数字を使用し、8 ~ 30 文字で構成される必要があります。

2.2 IP ホワイトリストの設定

背景

データベースのセキュリティおよび安定性を確保するため、ApsaraDB for Redis を使用する前に、データベースへのアクセスに使用する IP アドレスまたは IP セグメントをターゲットインスタンスのホワイトリストに追加する必要があります。ホワイトリストを正しく使用することで、ApsaraDB for Redis のアクセスセキュリティ保護が強化されます。ホワイトリストを定期的に管理することを推奨します。

前提条件

ホワイトリスト機能の正しい動作は、特定のカーネルバージョンでのみサポートされます。インスタンスのバージョンが最新でない場合は、ホワイトリストの設定中にメッセージが表示されます。ミラーバージョンのアップグレードについては、[マイナーバージョンのアップグレード](#)をご参照ください。

手順

1. [Redis コンソール](#)にログインして、ターゲットインスタンスを見つけます。
2. インスタンス ID または [管理] をクリックして、[インスタンス情報] ページに移動します。
3. 左側のナビゲーションウィンドウから [セキュリティ設定] を選択し、[デフォルトホワイトリストグループの変更] をクリックします。



注:

Redis インスタンスにカスタムホワイトリストグループを追加する場合は、[ホワイトリストグループの追加] をクリックします。カスタムホワイトリストの設定手順は、以下の手順と同じです。

4. [ホワイトリストグループの変更] ウィンドウで、"グループ名" と "イントラグループホワイトリスト" を入力し、[OK] をクリックします。IP リストで、[確認] をクリックします。

パラメーターの説明

- ・ グループ名は、小文字、数字、またはアンダースコアを使用し、2～32 文字で構成される必要があります。グループ名は小文字で始まり、文字または数字で終わる必要があります。ホワイトリストグループが正常に作成された場合は、この名前は変更不可です。
- ・ ホワイトリストには Redis インスタンスへのアクセスに使用するカスタム IP アドレスまたは IP セグメントを入力します。すべての IP アドレスからデータベースにアクセスするには、ホワイトリストを 0.0.0.0/0 に設定します。また、すべての IP アドレスからのデータベースへのアクセスを無効にするには、ホワイトリストを 127.0.0.1 に設定します。
 - 10.10.10.0/24 などの IP セグメントを入力した場合は、10.10.10.X という形式のすべての IP アドレスから、Redis インスタンスにアクセスすることを示します。
 - 複数の IP アドレスまたは IP セグメントを入力するには、コンマで区切ります。コンマの前後にはスペースを追加しないでください。
 - 1つのインスタンスに対して、最大 1,000 個の IP アドレスまたは IP セグメントを設定できます。

2.3 設定の変更

Alibaba Cloud ApsaraDB for Redis では、月払いまたは年払いと、フロー課金の 2 種類の課金方法があります。フロー課金は、月払いまたは年払いに変更可能です。2つのモードの設定は変更可能です。

背景

インスタンス設定を変更すると料金を変更されます。通常の課金については、「[ApsaraDB for Redis の価格](#)」をご参照ください。

注

- ・ 従量課金インスタンスおよびサブスクリプションインスタンスの両方で、リアルタイムでの設定変更がサポートされています。
- ・ クラスターインスタンスと非クラスターインスタンスは互いに切り替えられます。
- ・ 設定の変更中に、インスタンスは数秒間切断されることがあります。オフピーク時に設定を変更してください。

従量課金インスタンスの設定の変更

1. [Redis コンソール](#)にログインします。
2. インスタンスリストから対象のインスタンスに移動します。"操作"の[設定の変更]をクリックします。
3. アップグレードページで、設定を変更し、[有効化]をクリックします。

再設定が正常に完了したことがユーザーに通知され、新しい設定はすぐに有効になります。
(すぐに新しいモードで支払えます)

サブスクリプションインスタンスの設定の変更

1. [Redis コンソール](#)にログインします。
2. インスタンスリストでターゲットインスタンスに移動します。"操作"の[設定の変更]または[ダウングレード]をクリックします。
3. アップグレードページで、設定を変更し、[支払い]をクリックします。
4. 注文情報を確認して、[支払い]をクリックします。

手順は以上です。

2.4 ネットワークタイプの設定

背景

ApsaraDB では、クラシックと仮想プライベートクラウド (Virtual Private Cloud: VPC) の、2種類のネットワークタイプがサポートされています。

- ・ クラシックネットワーク: クラシックネットワーク上のクラウドサービスは隔離されず、クラウドサービスへの不正なアクセスはセキュリティグループまたはサービスのホワイトリストポリシーによってのみブロックされます。

- ・ VPC: Alibaba Cloud 上に隔離されたネットワーク環境を構築するのに役立ちます。VPC 上でルートテーブル、IP アドレス範囲、およびゲートウェイをカスタマイズします。さらに、専用線または VPN を介して、マシンルームと Alibaba Cloud VPC のクラウドリソースを仮想マシンルームに統合し、アプリケーションをクラウドへ円滑に移行します。

注

- ・ クラシックネットワークは VPC に切り替えられますが、その逆はできません。
- ・ クラシックネットワークを VPC に切り替えるときは、クラシックネットワークの IP アドレスを保持するかどうかを選択します。

前提条件

クラシックネットワークタイプの Redis インスタンスを VPC ネットワークタイプに切り替えるには、まず切り替えるインスタンスと同じリージョンに、VPC およびスイッチを作成します。



スイッチを作成するときは、Redis インスタンスと同じゾーンを選択します。

手順

1. [Redis コンソール](#)にログインし、ターゲットインスタンスを見つけて [管理] をクリックします。
2. インスタンス情報ページで [VPC へ切り替え] をクリックします。
3. [VPC へ切り替え] ページで、[VPC] および [VSwitch] を選択します。[クラシックネットワークの IP アドレスを保持する] および [保持期間] を選択し、[OK] をクリックします。



インスタンス情報ページで、[更新] をクリックしてアクセスアドレスを表示します。

2.5 インスタンスの更新

背景

- ・ 月払いまたは年払いのインスタンスは、契約期間内または有効期限後 7 日以内に更新します。時間ベースで支払うインスタンスはユーザーの実際の仕様に基づいているため、インスタンスの更新は不要です。詳しい情報については、「[延滞リマインダーおよび停止ポリシー](#)」をご参照ください。
- ・ ApsaraDB for Redis には、自動更新と手動更新の両方があります。

手順 (自動更新)

1. [Redis コンソール](#)にログインします。
2. [課金管理] > [更新] の順に選択し、更新ページへ移動します。
3. 左側のナビゲーションウィンドウで、[ApsaraDB for Redis] を選択します。
4. インスタンスの右側の [自動更新の有効化] をクリックし、[自動更新サイクル] を設定すると、自動更新が有効化されます。

手順 (手動更新)



:

手動での更新時にインスタンスの設定をリセットすることができます。更新せずに、設定を変更する必要がある場合は、「[設定の変更](#)」をご参照ください。

1. [Redis コンソール](#)にログインし、更新するインスタンスを見つけて[更新] をクリックします。
2. 注文の確認ページに移動し、予定される更新期間を選択して [OK] をクリックします。



注:

インスタンスの設定をリセットするには、[設定の変更] を選択し、ターゲット構成を設定します。設定変更の情報と更新期間中の課金基準については、[スペック変更](#) をご参照ください。

3. 送信後に支払いページに移動して、希望する支払方法を選択し、[サブスクリプションの確認] をクリックします。

最後に、支払いが正常に完了したことを確認するページにリダイレクトされます。

2.6 メンテナンス時間の設定

背景

ApsaraDB for Redis インスタンスの安定性を確保するために、バックエンドシステムでは Alibaba Cloud プラットフォーム上のインスタンスとマシンが不定期にメンテナンスされます。

公式メンテナンスの前に、ApsaraDB for Redis から Alibaba Cloud アカウントに設定された連絡先に、SMS メッセージとメールが送信されます。

メンテナンスプロセス中の安定性を確保するため、事前に指定されたメンテナンス日の運用時間より前に、インスタンスはメンテナンス中 状態に入ります。インスタンスがメンテナンス中状態のときでも、データベースのデータへの通常のアクセスには影響はありません。しかし、変更

関連した機能 (たとえば、設定変更など) は、コンソール上のこのインスタンスでは一時的に利用不可になります。一方、パフォーマンスモニタリングなどの照会機能は引き続き利用できます。



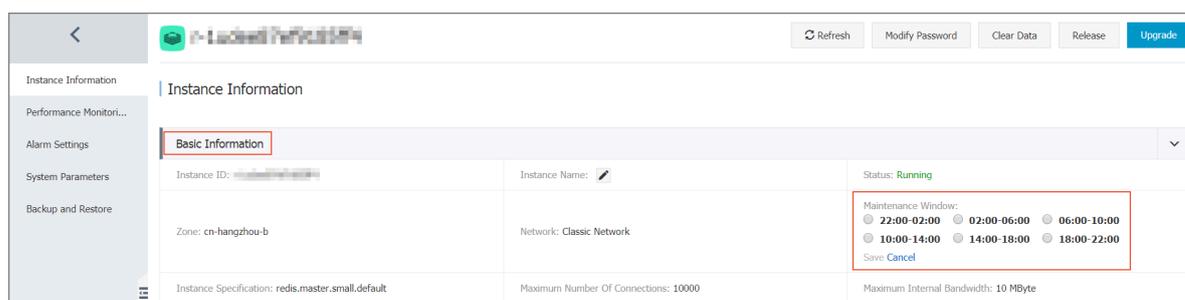
:

事前に設定されたメンテナンス時間になると、メンテナンス中にインスタンスが断続的に中断される可能性があるため、可能であれば、オフピーク時間帯にインスタンスをメンテナンスすることを推奨します。

手順

1. [Redis コンソール](#)にログインし、ターゲットインスタンスを見つけます。
2. [インスタンス ID] または [管理] をクリックして、インスタンス情報ページに移動します。
3. 次の図に示すように、[基本情報] の [メンテナンス時間] の隣の [設定] をクリックします。

ApsaraDB for Redis のデフォルトのメンテナンス時間は、午前 2 時から午前 6 時です。



4. メンテナンス時間を選択し、[保存] をクリックします。



注:

時間帯は北京時間です。

2.7 マイナーバージョンのアップグレード

背景

ApsaraDB for Redis では、セキュリティの脆弱性の修正とサービスの安定性向上のために、重大なカーネルの最適化が実行されます。コンソール上でワンクリックすると、最新のカーネルバージョンにアップグレードされます。



:

- ・ システムではインスタンスのカーネルバージョンが自動的に検出されます。現在のバージョンが最新の場合は、コンソールの基本情報ページに [マイナーバージョンのアップグレード] ボタンは表示されません。

- ・ カーネルバージョンをアップグレードすると接続が 30 秒間中断されるので、オフピーク時間にアップグレードを実行し、アプリケーションに再接続メカニズムがあることを確認します。

手順

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページに移動します。
3. 基本情報の [マイナーバージョンのアップグレード] をクリックします。
4. [マイナーバージョンのアップグレード] ウィンドウで [今すぐアップグレード] をクリックします。

基本情報ページで、インスタンスのステータスに [マイナーバージョンのアップグレード中...] と表示されます。インスタンスのステータスに "In Use" と表示されたら、アップグレードは完了です。

2.8 データの消去



:

データのクリーンアップ操作では、対象のインスタンスのすべてのデータが消去され、消去されたデータは復元されません。

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページへ移動します。
3. [データの消去] をクリックし、表示された確認ダイアログで [OK] をクリックします。
4. 携帯電話の確認ページで、確認コードを入手して入力し、データのクリーンアップを完了します。

2.9 インスタンスのリリース



注:

従量課金インスタンスは随時リリースできますが、サブスクリプションインスタンスは手動で削除またはリリースすることができません。

手順

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. [インスタンス ID] または [管理] をクリックし、インスタンス情報ページに移動します。

3. [リリース] をクリックし、ポップアップウィンドウで [OK] をクリックします。
4. 携帯電話の確認ページで、確認コードを入手して入力し、インスタンスをリリースします。

3 接続管理

3.1 クラシックネットワークの保持期間の変更

クラシックネットワークのアクセスアドレス保持期間を設定した後、期限が切れる前にコンソールで保持期間を延長します。

ハイブリッドアクセス期間中は、必要に応じていつでも元のクラシックネットワークの保持期間を変更できます。有効期限は新しい日付から再計算されます。たとえば、元のクラシックネットワークのイントラネットアドレスの期限が2017年8月18日に設定されている場合、2017年8月15日に有効期限を14日後に変更すると、イントラネットアドレスは2017年8月29日にリリースされます。

手順

1. [Redis コンソール](#)にログインします。
2. インスタンスリストページで、ターゲットインスタンスを見つけて [管理] をクリックし、[インスタンス情報] ページに移動します。
3. [クラシックネットワークの接続アドレスの保持] エリアで、[保持期間の変更] をクリックします。
4. 表示されたダイアログボックスで、新しい有効期限を選択し [OK] をクリックします。

4 バックアップとリカバリ

4.1 バックアップとリカバリ

究極の永続的ストレージエンジンとして、ApsaraDB for Redis がますます多くの業務で使用されるにつれて、ユーザーからはデータの高い信頼性が求められています。ApsaraDB for Redis のバックアップとリカバリソリューションによって、総合的なデータの信頼性のアップグレードが可能になります。

バックアップとリカバリの詳細については、以下のビデオをご覧ください。ビデオは約3分です。

<https://cloud.video.taobao.com/play/u/3050941791/p/1/e/6/t/1/56540476.mp4>

自動バックアップ (バックアップポリシー設定)

背景

ますます多くのアプリケーションによって永続ストレージとして ApsaraDB for Redis が使用されるので、操作を誤った場合には、迅速にデータを復元するために従来のバックアップメカニズムが必要です。Alibaba Cloud では、スレーブノードで RDB のスナップショットのバックアップが実行され、バックアッププロセスの間、インスタンスのパフォーマンスが保護されます。Alibaba Cloud は便利なコンソールを提供しており、バックアップ設定をカスタマイズできます。

手順

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページに移動します。
3. 左側のナビゲーションウィンドウで [バックアップとリカバリ] を選択します。
4. [バックアップ設定] をクリックします。
5. [編集] をクリックし、自動バックアップサイクルおよび自動バックアップ時間をカスタマイズします。



デフォルトでは、バックアップデータは7日間保持されます。この設定は変更できません。

6. [OK] をクリックし、自動バックアップ設定を完了します。

手動バックアップ (即時バックアップ)

一般的なバックアップの設定に加えて、いつでもコンソールで手動バックアップリクエストを開始できます。

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページに移動します。
3. 左側のナビゲーションウィンドウで [バックアップとリカバリ] を選択します。
4. 右上にある [作成] をクリックします。
5. [OK] をクリックすると、インスタンスは直ちにバックアップされます。



:

バックアップデータ ページで、期間を選択し、バックアップデータの履歴を照会します。デフォルトでは、バックアップデータは 7 日間保持されるので、過去 7 日間のバックアップデータの履歴が照会されます。

バックアップアーカイブ

背景

業界の規制または企業のポリシーの要件によって、Redis データのバックアップを定期的にアーカイブする必要があります。ApsaraDB for Redis では現在無料でバックアップのアーカイブ機能が提供され、自動および手動バックアップファイルが OSS へ保存されます。現在、Alibaba Cloud では、バックアップアーカイブは無料で 7 日間 OSS に保存されます。7 日後、ファイルは自動的に削除されます。

バックアップアーカイブをより長期間保持する場合は、コンソール上のリンクをコピーして、データベースバックアップファイルをローカルストレージに手動でダウンロードします。

手順

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページに移動します。
3. 左側のナビゲーションウィンドウで [バックアップとリカバリ] を選択します。
4. バックアップデータページで、アーカイブされたバックアップデータを選択し、[ダウンロード] をクリックします。

データのりカバリ

データリカバリ機能によって、データベースの誤操作による損害が最小限に抑えられます。現在、ApsaraDB for Redis では、バックアップからのリカバリがサポートされています。

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報 ページに移動します。
3. 左側のナビゲーションウィンドウで [バックアップとリカバリ] を選択します。
4. [バックアップとリカバリ] ページで [バックアップデータ] タブをクリックします。
5. リカバリする期間を選択し、[検索] をクリックします。対象のバックアップファイルを選択し、[データのリカバリ] をクリックします。
6. [データリカバリ] ウィンドウで、[OK] をクリックし、データを元のインスタンスに直接リカバリします。他に、[インスタンスのクローン] を選択し、バックアップデータを新しいインスタンスにリカバリする方法もあります。リカバリされたデータが正しいことを確認したら、データを元のインスタンスにリカバリします。



:

データリカバリ 操作は非常にリスクが高く、時間が許す場合はインスタンスのクローンを作成することを推奨します。この方法では、リカバリするバックアップデータセットに基づいて従量課金インスタンスを作成します。データが正しいことを確認したら、データを元のインスタンスにリカバリします。

インスタンスのクローン

背景

日常的なメンテナンスプロジェクト中、運用エンジニアは新しいアプリケーションを迅速にデプロイする必要があることが多いです。アプリケーションのデプロイメントが比較的単純な場合は、新しいインスタンスは ECS イメージファイルに基づいて簡単に作成されます。データベースレベルでは、デプロイメントはより複雑になります。運用エンジニアは新しいデータベースを購入またはインストールし、関連するデータベーススクリプト (テーブル、トリガー、およびビューを作成するために) を初期化します。このようなシナリオでは、多くの細かい操作が実行され、エラー率が比較的高くなります。特にサービスのアクティベーションが早いゲーム業界では、新しいアプリケーションの迅速なデプロイメントが毎日何度も繰り返されます。

この問題点に対処するため、ApsaraDB for Redis ではインスタンスのクローン作成機能が開発され、バックアップファイルから新しいサブスクリプションインスタンスまたは従量課金インスタンスに迅速に複製されます。グラフィカルインターフェイスでシングルクリックするだけで、データベース開発とデプロイメントの複雑な操作が実行され、生産性が大幅に向上します。

手順

1. [Redis コンソール](#)にログインし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、インスタンス情報ページに移動します。

3. 左側のナビゲーションウィンドウで [バックアップとリカバリ] を選択します。
4. バックアップデータページで、複製するバックアップデータセットを選択し、[インスタンスのクローン] をクリックします。

5 データ移行

5.1 オンプレミス Redis から ApsaraDB Redis への移行

5.1.1 オンプレミス Redis クラスターから ApsaraDB Redis クラスターへのデータ移行

redis-sync-manager を使用して、オンプレミス Redis クラスターから ApsaraDB Redis クラスターへデータを移行します。

前提条件

- ・ redis-sync-manager では、[redis-port を使用したデータの移行](#)を行うため、`$PATH` 環境変数で redis-port を指定します。
- ・ ソースとターゲットの基本データに必要なメモリ使用量の要件、および現在の並行性について知っておく必要があります。
- ・ ソースクラスター内のどのスロットも中間移行状態になっていないことを確認します。

移行ツール

- ・ redis-port: シングル Redis プロセスからターゲット ApsaraDB for Redis クラスターにデータを同期するために使用されます。
- ・ redis-sync-manager : redis-port の補助として機能し (redis-portは redis-sync-manager にカプセル化されています)、ソース Redis クラスターからターゲット ApsaraDB for Redis クラスターにデータを同期するために使用されます。

ツールのダウンロード

[redis-sync-manager](#)

説明

次のコマンドを実行して、ユーザーが作成した Redis クラスターから ApsaraDB for Redis クラスターにデータを移行します。

```
./ redis - sync - manager -- from = src_host : src_port -- target = dst_host : dst_port [-- password = src_passwo rd ] [-- auth = dst_passwo rd ] [-- filterkey = " str1 | str2 | str3 " ] [-- targetdb = DB ] [-- rewrite ] [-- bigkeysize = SIZE ] [-- logfile = REDISPORT . LOG ] [-- httpport = HTTPPORT ] [-- sync - parallel = INT ] [-- sync - role = " master | slave " ]
```

redis-sync-manager の移行メカニズム

redis-sync-manager は `src_host : src_port` と対話して、最初に `cluster nodes` コマンドを通じてクラスタートポロジに関する情報を取得します。次に、`-- sync - role` パラメーター設定に基づいて、同期されるシャードの `IP : PORT` 一覧が返されます。最後に、redis-sync-manager は `redis-port` を呼び出してデータを同期させます。(完全データ同期の並行性は `-- sync - parallel` パラメーター設定によって決まり、ソースクラスター内の各シャードの増分データはターゲットクラスター内の対応するシャードに同期されます)。次の表で、関連パラメーターについて説明します。

表 5-1: パラメーターの説明

パラメーター	説明
<code>src_host</code>	ユーザーが作成した Redis データベースのドメイン名または IP アドレス。このパラメーターをユーザーが作成した Redis クラスター内のプロセスの IP アドレスに設定します。
<code>src_port</code>	ユーザーが作成した Redis データベースのポート。このパラメーターを <code>src_host</code> に指定された IP アドレスに対応するポートに設定します。
<code>src_password</code>	ユーザーが作成した Redis データベースのパスワード
<code>dst_host</code>	ApsaraDB for Redis データベースのドメイン名
<code>dst_port</code>	ApsaraDB for Redis データベースのポート
<code>dst_password</code>	ApsaraDB for Redis データベースのパスワード
<code>str1 str2 str3</code>	<code>str1</code> 、 <code>str2</code> 、または <code>str3</code> でキーをフィルターします。
<code>DB</code>	ApsaraDB for Redis に同期するデータベース
<code>rewrite</code>	すでに書き込まれているキーを上書きします。
<code>bigkeysize=SIZE</code>	書き込まれた値が <code>SIZE</code> より大きい場合、ビッグキー書き込みモードを使用していることを示します。
<code>--logfile=REDISPORT.LOG</code>	<code>edis-sync-manager.log</code> などのログファイル名。移行中、移行されたシャードに対して異なるログファイルが生成され、移行中のシャードのシリアル番号が対応するログファイル名に自動的に追加されます。 デフォルト値: <code>logs / redis - sync - manager . log</code>
<code>--sync-parallel=INT</code>	並行性がデータ同期でサポートされているかどうかを示し、実行可能なメモリ使用量を表示します。デフォルト値: <code>1</code>
<code>--sync-role="master slave"</code>	ソースクラスターのプライマリデータベースまたはセカンダリデータベースの同期順序を示します。デフォルト値: <code>master</code>

移行例

```

$redis-sync-manager --from=127.0.0.1:18380 --target=127.0.0.1:18385 --ncpu=2 --sync-role=slave --rewrite --sync-parallel=1 --auth= --password= --logfile=sync.log --pidfile=sync.pid
2018/10/24 14:42:29
changed param:
--from=127.0.0.1:18382
--logfile=sync.log.0-5460
--pidfile=sync.pid.0-5460
--httpport=16000
cmd:
redis-port sync --from=127.0.0.1:18382 --password= --ncpu=2 --auth= --logfile=sync.log.0-5460 --pidfile=sync.pid.0-5460 --target=127.0.0.1:18385 --httpport=16000 --rewrite
2018/10/24 14:42:50 [FROM:127.0.0.1:18382]sync rdb done

2018/10/24 14:42:50
changed param:
--httpport=16001
--from=127.0.0.1:18383
--logfile=sync.log.5461-10922
--pidfile=sync.pid.5461-10922
cmd:
redis-port sync --target=127.0.0.1:18385 --httpport=16001 --rewrite --from=127.0.0.1:18383 --password= --ncpu=2 --auth= --logfile=sync.log.5461-10922 --pidfile=sync.pid.5461-10922
2018/10/24 14:43:15 [FROM:127.0.0.1:18383]sync rdb done

2018/10/24 14:43:15
changed param:
--from=127.0.0.1:18384
--logfile=sync.log.10923-16383
--pidfile=sync.pid.10923-16383
--httpport=16002
cmd:
redis-port sync --from=127.0.0.1:18384 --password= --ncpu=2 --auth= --logfile=sync.log.10923-16383 --pidfile=sync.pid.10923-16383 --target=127.0.0.1:18385 --httpport=16002 --rewrite
2018/10/24 14:43:38 [FROM:127.0.0.1:18384]sync rdb done

2018/10/24 14:43:38 127.0.0.1:18382 is sync is finished, syncing incremental data
2018/10/24 14:43:38 127.0.0.1:18383 is sync is finished, syncing incremental data
2018/10/24 14:43:38 127.0.0.1:18384 is sync is finished, syncing incremental data
2018/10/24 14:43:38 there are 3 shard needed to sync, 3 is finished , 0 is failed. please check...

```

画像のコールアウト部分を以下に説明します。

1. 各シャードの同期情報と同期ステータスを表示します。
2. IP : PORT は `-- sync - role` パラメーター設定に基づいて、シャードごとに選択され、IP:PORT を1つずつプリントアウトします。
3. 各シャードの同期ステータスをログファイルに出力します。

5.1.2 redis-port を使用したデータの移行

redis-port を使用して、自己管理 Redis データベースから ApsaraDB for Redis インスタンスにデータを移行します。

前提条件

- ・ 対象の ApsaraDB for Redis インスタンスが存在する VPC に、Linux ベースの ECS (Elastic Compute Service) インスタンスが作成済みである必要があります。
- ・ 上記の ECS インスタンスに `redis-port` がダウンロードされている必要があります。
- ・ `chmod u + x redis - port` を実行して、`redis-port` を実行可能ファイルに変更している必要があります。

手順

1. ECS インスタンスに組み込まれている Linux システムにログインします。
2. `redis-port` が存在するディレクトリで、以下のコードを実行して移行を開始します。

```

./ redis - port sync -- from = src_host : src_port -- Password
= src_passwo rd -- target = dst_host : dst_port -- auth

```

```
= dst_passwo rd [-- filterkey =" str1 | str2 | str3 "] [--
targetdb = dB ] [-- rewrite ] [-- bigkeysize = size ] [-- logfile =
redisport . log ]
```

表 5-2 : redis-port の引数

引数	説明
src_host	自己管理 Redis データベースのドメイン名 (または IP)
src_port	自己管理 Redis データベースのポート
src_password	自己管理 Redis データベースのパスワード
dst_host	ApsaraDB for Redis インスタンスのドメイン名
dst_port	ApsaraDB for Redis インスタンスのポート
dst_password	ApsaraDB for Redis インスタンスのパスワード
str1 str2 str3	str1、str2、または str3 のフィルターキー
DB	移行する自己管理 Redis DB のインデックス
rewrite	ApsaraDB for Redis インスタンスにすでに存在する同一のキーを上書きする  : この引数が設定されておらず、両方のデータベースに同一キーが存在する場合、移行は正しく実行されない可能性があります。
bigkeysize	size の値よりも大きいキーを書き込むときは、ビッグキー書き込みを有効にします。
logfile	ログを保存するファイルを指定します。

3. ログをモニタリングして、移行手順が期待どおりに実行されることを確認します。

```
[root@redisTest ~]# ./redis-port sync --from=127.0.0.1:6379 --password= --target=r- .redis.rds.aliyuncs.com:6379 --auth=
2018/12/06 17:34:47 [INFO] set ncpu = 2, parallel = 2 filterdb = 0 targetdb = -1
2018/12/06 17:34:47 [INFO] set ncpu = 2, parallel = 2 filterdb = 0 targetdb = -1
2018/12/06 17:34:47 [INFO] sync from '127.0.0.1:6379' to 'r- .redis.rds.aliyuncs.com:6379' http '%!(int=0)'
2018/12/06 17:34:47 [INFO] sync from '127.0.0.1:6379' to 'r- .redis.rds.aliyuncs.com:6379'
2018/12/06 17:34:47 [INFO] rdb file = 1124803
2018/12/06 17:34:47 [INFO] Aux information key:redis-ver value:3.2.12
2018/12/06 17:34:47 [INFO] Aux information key:redis-bits value:64
2018/12/06 17:34:47 [INFO] Aux information key:ctime value:1544088887
2018/12/06 17:34:47 [INFO] Aux information key:used-mem value:3708224
2018/12/06 17:34:47 [INFO] db_size:5000 expire_size:0
2018/12/06 17:34:48 [INFO] total=1124803 - 441129 [ 39%] entry=1919
2018/12/06 17:34:49 [INFO] total=1124803 - 868155 [ 77%] entry=3813
2018/12/06 17:34:50 [INFO] total=1124803 - 1124803 [100%] entry=5000
2018/12/06 17:34:50 [INFO] sync rdb done
2018/12/06 17:34:51 [INFO] sync: +forward=1 +nbypass=0 +nbytes=14
2018/12/06 17:34:52 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:53 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:54 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:55 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:56 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
```



注:

- ・ `sync rdb done` 後にログを確認します。+ `forward = 1` を含むエントリの + `nbytes` 値が 14 より大きい場合は、増分同期ログです。増分同期ステータスをモニタリングして、データベースの最適な切り替え時期を判断します。
- ・ 完全同期後、同期ソース (つまり自己管理 Redis データベース) は定期的な `ping` リクエストを送信し、+ `forward = 1` と + `nbytes = 14` を持つログを生成します。これらは増分同期ログではありません。

5.1.3 Data Transmission Service (DTS) を使用したデータの移行

DTS を使用して、外部 Redis インスタンスから ApsaraDB for Redis インスタンスにデータを移行します。

外部 Redis インスタンスから ApsaraDB for Redis インスタンスへのデータの移行

DTS を使用して、2 つの Redis インスタンス間でデータを移行できます。ソースインスタンスが、増分データ同期をサポートしている外部 Redis インスタンスの場合、データ移行はローカルアプリケーションの書き込み操作に影響しません。

本ページでは、DTS を使用して外部 Redis インスタンスから ApsaraDB for Redis インスタンスにデータを移行する手順について説明します。同じ手順に従って、ApsaraDB for Redis インスタンスから別のインスタンスにデータを移行できます。

移行モード

ソースインスタンスが外部 Redis インスタンスの場合、標準移行と増分データ移行の両方がサポートされています。ソースインスタンスが ApsaraDB for Redis インスタンスの場合、標準移行のみがサポートされています。標準移行と増分データ移行には、次の機能と制限があります。

- ・ 標準移行

DTS は、外部 Redis インスタンス上の既存のキーをすべて ApsaraDB for Redis インスタンスに移行します。

- ・ 増分データ移行

増分データ移行中、外部 Redis インスタンスの更新されたキーは ApsaraDB for Redis インスタンスに同期されます。同期プロセス後、ApsaraDB for Redis インスタンスは外部 Redis インスタンスから動的にデータの複製を開始します。増分データ移行を使用することにより、外部 Redis インスタンスは ApsaraDB for Redis インスタンスにスムーズにデータを移行しながら通常のサービスを提供します。

移行機能

Redis 増分データ移行は、次のコマンドをサポートします。

- ・ APPEND
- ・ BITOP、BLPOP、BRPOP、BRPOPLPUSH
- ・ DECR、DECRBY、DEL
- ・ EVAL、EVALSHA、EXEC、EXPIRE、EXPIREAT
- ・ FLUSHALL、FLUSHDB
- ・ EOADD、GETSET
- ・ HDEL、HINCRBY、HINCRBYFLOAT、HMSET、HSET、HSETNX
- ・ INCR、INCRBY、INCRBYFLOAT
- ・ LINSERT、LPOP、LPUSH、LPUSHX、LREM、LSET、LTRIM
- ・ MOVE、MSET、MSETNX、MULTI
- ・ PERSIST、PEXPIRE、PEXPIREAT、PFADD、PFMERGE、PSETEX、PUBLISH
- ・ RENAME、RENAMENX、RESTORE、RPOP、RPOPLPUSH、RPUSH、RPUSHX
- ・ SADD、SDIFFSTORE、SELECT、SET、SETBIT、SETEX、SETNX、SETRANGE、SINTERSTORE、SMOVE、SPOP、SREM、SUNIONSTORE
- ・ ZADD、ZINCRBY、ZINTERSTORE、ZREM、ZREMRANGEBYLEX、ZUNIONSTORE、ZREMRANGEBYRANK、ZREMRANGEBYSCORE

前提条件

ソース Redis インスタンスが、専用回線を介して Alibaba Cloud VPC に接続された外部 Redis インスタンスか、または VPC 内の ApsaraDB for Redis インスタンスである場合、データ転送にプロキシが必要です。

専用回線を介して DTS が VPC 接続 Redis インスタンスまたは Alibaba Cloud に接続された外部 Redis インスタンスに接続できるようにするには、EIP が割り当てられた ECS インスタンスを選択し、Nginx を ECS インスタンスにデプロイして Nginx プロキシ転送を設定する必要があります。

Redis インスタンスと ECS インスタンスは同じ VPC に接続する必要があります。専用回線を介して Alibaba Cloud に接続されている外部 Redis インスタンスの場合、ECS インスタンスが存在する VPC は Alibaba Cloud VPC である必要があります。

次のセクションでは、DTS サーバーが VPC 接続 Redis インスタンスに接続可能となるように、Nginx を使用して転送を設定する方法を説明します。

1. Nginx のデプロイ

ECS インスタンスで次のコマンドを実行して Nginx をデプロイします。

```
yum install nginx
```

2. Nginx 転送の設定

Nginx をインストールしたら、設定ファイル `/etc/nginx/nginx.conf` を変更し、バックエンドを設定して Redis インスタンスをリスンします。設定ファイル内の HTTP 設定をコメントアウトし、TCP 設定を追加します。コメントアウトする HTTP 設定は次のとおりです。

```
# {
# log_format main '$remote_addr - $remote_user [$time_local] "$request" '
# '$status $body_bytes_sent "$http_referer" '
# '"$http_user_agent" "$http_x_forwarded_for"';
#
# access_log /var/log/nginx/access.log main;
#
# sendfile on;
# tcp_nopush on;
# tcp_nodelay on;
# keepalive_timeout 65;
# types_hash_max_size 2048;
#
# include /etc/nginx/mime.types;
# default_type application/octet-stream;
#
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
# include /etc/nginx/conf.d/*.conf;
#}
```

次の TCP 設定を設定ファイルに追加します。

```
stream {
    upstream backend {
        hash $ remote_add r consistent ;
        # Set the connection string and port for
        connecting the backend server to the Redis instance
        . The connection timeout period is 10 seconds and
        the maximum number of attempts is three .
        server r - bp1b294374 634044 . redis . rds . aliyuncs .
        com : 6379 max_fails = 3 fail_timeo ut = 10s ;
    }
    server {
        # Nginx connection port
        Listen 3333 ;
        # Set the timeout period to 20 seconds for
        connecting Nginx to the backend server .
        proxy_conn ect_timeou t 20s ;
        # If five minutes have passed since the last
        successful access ( connection or read / write ) to the
        backend server , the connection is timed out and
        disconnect ed .
        proxy_time out 5m ;
        # Redirect the TCP connection , data reception ,
        and data forwarding to backend server " backend " .
        proxy_pass backend ;
    }
}
```

```
}
```

たとえば、接続する Redis インスタンスの接続文字列は `r - bp1b294374 634044 . redis . rds . aliyuncs . com : 6379` であり、Nginx 接続ポートは 3333 です。TCP 設定を次のように設定します。

```
stream{
    upstream backend{
        hash $remote_addr consistent;
        server r-XXXXXXXXXXXX.redis.rds.aliyuncs.com:6379 max_fails=3 fail_timeout=10;
    }
    server{
        listen 3333;
        proxy_connect_timeout 20s;
        proxy_timeout 5m;
        proxy_pass backend;
    }
}
```

3. Nginx 接続ポートを介した Redis インスタンスへの接続

上記の設定が完了したら、Nginx を実行して Nginx プロキシサービスを開始します。

Nginx がデプロイされている ECS インスタンスの EIP が `114 . 55 . 89 . 152` の場合、`redis_cli` を使用して接続ポートに直接接続し、プロキシサービスが正常に動作しているかどうかをテストします。

```
[root@iZ23bsclllgZ ~]# redis-cli -h 114.55.89.152 -p 3333
114.55.89.152:3333> auth XXXXXXXXXXXX
OK
114.55.89.152:3333> ping
PONG
114.55.89.152:3333> get a1
"10"
```

前述の図に示すように、Nginx 接続ポートを介して Redis インスタンスに接続します。

DTS に移行タスクを設定する際、ソースインスタンスが VPC 接続 Redis インスタンスまたは専用回線を介して Alibaba Cloud に接続された外部 Redis インスタンスである場合は、Nginx プロキシ接続アドレスを直接指定します。

DTS が VPC をサポートするようになった場合、VPC 接続 Redis インスタンスまたは専用回線を介して Alibaba Cloud に接続された外部 Redis インスタンスにプロキシサービスは必要ありません。DTS が VPC をサポート可能なスケジュールについては、Alibaba Cloud Web サイトのお知らせをご参照ください。

移行タスクの設定

前提条件がすべて満たされたら、移行タスクの設定を開始します。この例では、Alibaba Cloud に接続された外部 Redis インスタンスから、専用回線を介してクラシックネットワーク内の ApsaraDB for Redis インスタンスにデータを移行する方法を説明します。

1. [DTS コンソール](#) にログインし、右上隅の [移行タスクの作成] をクリックして移行タスクを設定します。

2. インスタンス接続情報の入力

この手順では、タスク名、外部 Redis インスタンスへの接続情報、および ApsaraDB for Redis インスタンスへの接続情報を入力する必要があります。接続情報には、以下のパラメーターが含まれます。

- ・ タスク名

DTS は各タスクのタスク名を自動的に生成します。タスク名は一意である必要はありません。必要に応じてタスク名を変更します。業務に適したタスク名を選択することを推奨します。

- ・ ソースインスタンス情報

インスタンスタイプ: EIP が割り当てられた外部 Redis インスタンスを選択します。

インスタンスリージョン: 外部 Redis インスタンスの場合、インスタンスに地理的に最も近いリージョンを選択します。リージョンが Redis インスタンスに近いほど、移行パフォーマンスが向上します。

データベース種類: Redis を選択します。

インスタンスモード: デフォルトでモードをスタンドアロンに設定します。Redis クラスターのサポートはまもなく利用可能になります。

ホスト名または IP アドレス: 外部 Redis インスタンスの接続文字列。Nginx プロキシが設定されている場合は、このパラメーターを Nginx への接続に使用されるアドレスに設定します。

ポート: 外部 Redis インスタンスのリスニングポート。Nginx プロキシが設定されている場合は、このパラメーターを Nginx ポートに設定します。

データベースパスワード: 外部 Redis インスタンスへの接続に使用されるパスワード。このパラメーターは必須ではありません。外部 Redis インスタンスにパスワードがない場合、このパラメーターは空のままにします。

- ・ ターゲットインスタンス情報

インスタンスタイプ: Redis インスタンス。

インスタンスリージョン: ApsaraDB for Redis インスタンスを作成したリージョン。

Redis インスタンス ID: データの移行先となるターゲット ApsaraDB for Redis インスタンスの ID。

データベースパスワード: ApsaraDB for Redis インスタンスへの接続に使用されるパスワード。

3. 接続情報を設定したら、右下隅にある [ホワイトリストを許可して次の手順に進む] をクリックして移行オブジェクトを選択します。

4. 移行オブジェクトと移行タイプの選択

この手順では、移行オブジェクトと移行タイプを設定する必要があります。

・ 移行タイプ

Redis インスタンスの場合、DTS は標準移行と増分データ移行をサポートします。

データをすべて移行する場合は、移行タイプとして標準移行を選択します。

ソースインスタンスへの書き込み操作に影響を与えることなくデータを移行する必要がある場合は、移行タイプとして標準移行 + 増分移行を選択します。

・ 移行オブジェクト

この手順では、移行するデータベースを選択する必要があります。現在、Redis はデータベース全体の移行のみをサポートしています。キーを選択する代わりにデータベースを選択する必要があります。

5. プレチェック

移行タスク開始前にプレチェックが必要です。移行タスクは、プレチェックに合格した後のみ開始できます。プレチェックの詳細については、「[プレチェック](#)」をご参照ください。

プレチェックが失敗した場合は、チェックリストの特定のチェック項目の右側にあるボタンをクリックして失敗の詳細を確認し、問題をトラブルシューティングし、プレチェックを再実行します。

6. 移行タスクの開始

プレチェックに合格したら、移行タスクを開始します。タスクが正常に開始されたら、タスクリストで移行ステータスと進行状況を確認します。

上記の操作が完了したら、外部 Redis インスタンスから ApsaraDB for Redis インスタンスにデータを移行するための移行タスクが正常に設定されています。

プレチェック

DTS は、移行を開始する前にプレチェックを実行します。このセクションでは、プレチェック項目を説明します。

チェック項目	説明	備考
ソースデータベース接続	DTS サーバーと外部 Redis インスタンス間の接続をチェックします	1. 入力した情報が正しいかどうかをチェックします。情報が正しくない場合は、情報を修正してプレチェックを再実行します。 2. ポートが他のサーバーから開始された接続を許可するかどうかをチェックします。
ターゲットデータベース接続	DTS サーバーとターゲット ApsaraDB for RDS インスタンス間の接続をチェックします	入力した情報が正しいかどうかをチェックします。正しくない場合は、情報を修正してプレチェックを再実行します。
データベースの 1 対 1 チェック	データが複数のデータベースから 1 つのデータベースに移行されているかどうかをチェックします	現在、DTS は複数のデータベースから 1 つのデータベースへの移行をサポートしていません。このような場合は、タスク設定を修正してプレチェックを再実行します。
ターゲットデータベースが空かどうかのチェック	ターゲット Redis インスタンス上のデータの移行先データベースが空かどうかをチェックします	このチェックが失敗した場合は、ターゲットデータベース内のキーをすべて削除し、プレチェックを再実行します。
増分移行タスクのトポロジ的な競合のチェック	他の増分移行タスクがターゲット Redis インスタンス上で実行されているかどうかをチェックします	このチェックが失敗した場合は、他の増分移行タスクを終了し、プレチェックを再実行します。

5.1.4 RDB ファイルを使用したオンプレミス Redis データの ApsaraDB Redis への移行

Redis-port ツールにより、RDB ファイルを使用してオンプレミス Redis データベースのデータを ApsaraDB Redis に同期させることが可能です。

Redis-port のダウンロード

Redis-port アドレス

例

```
./ redis - port restore -- input = x / dump . rdb -- target = dst_host : dst_port -- auth = dst_passwo rd [-- filterkey = " str1 | str2 | str3 " ] [-- targetdb = DB ] [-- rewrite ] [-- bigkeysize = SIZE ] [-- logfile = REDISPORT . LOG ]
```

パラメーターの説明

- ・ x/dump.rdb: ユーザーが作成した Redis データベースのダンプファイルパスを示します。

- ・ `dst_host`: ApsaraDB for Redis データベースのドメイン名を示します。
- ・ `dst_port`: ApsaraDB for Redis データベースのポートを示します。
- ・ `dst_password`: ApsaraDB for Redis データベースのパスワードを示します。
- ・ `str1|str2|str3`: キーを `str1`、`str2`、または `str3` でフィルターします。
- ・ `DB`: ApsaraDB for Redis に同期するデータベースを示します。
- ・ `rewrite`: すで書き込まれているキーを上書きします。
- ・ `bigkeysize=SIZE`: 書き込まれた値が `SIZE` よりも大きい場合、ビッグキー書き込みモードが使用されることを示します。

Redis-port ログに基づいてデータ同期ステータスを確認

```

$ /bin/redis-port restore --input=/... --target=... --auth=...
2018/01/24 10:31:20 [INFO] set ncpu = 24, parallel = 24, filterdb = 0, targetdb = 1
2018/01/24 10:31:20 [INFO] restore from '/home/pan.liang/run.4.0/dump_test.rdb' to '10.218.144.110:8179'
2018/01/24 10:31:20 [INFO] Aux information key:redis-ver value:4.0.2
2018/01/24 10:31:20 [INFO] Aux information key:redis-bits value:64
2018/01/24 10:31:20 [INFO] Aux information key:ctime value:1516359975
2018/01/24 10:31:20 [INFO] Aux information key:used-mem value:448742200
2018/01/24 10:31:20 [INFO] Aux information key:repl-stream-db value:0
2018/01/24 10:31:20 [INFO] Aux information key:repl-id value:e19e45132deb3be68a1e97a17fced710ff1e02
2018/01/24 10:31:20 [INFO] Aux information key:repl-offset value:46308728805
2018/01/24 10:31:20 [INFO] Aux information key:aof-preamble value:0
2018/01/24 10:31:20 [INFO] db_size:2977 expire_size:0
2018/01/24 10:31:21 [INFO] total = 177007551 - 40387120 [ 22%] entry=770
2018/01/24 10:31:22 [INFO] total = 177007551 - 86511868 [ 48%] entry=863
2018/01/24 10:31:23 [INFO] total = 177007551 - 126672277 [ 71%] entry=1472
2018/01/24 10:31:24 [INFO] total = 177007551 - 159105721 [ 89%] entry=1472
2018/01/24 10:31:25 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:26 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] restore: rdb done

```

`restore : rdb done` が表示された場合、データの同期は完了です。

5.1.5 AOF を使用したオンプレミス Redis データの ApsaraDB Redis への移行

Redis-cli ツールにより、AOF を使用してオンプレミス Redis データベースのデータを ApsaraDB Redis に移行可能です。

Redis-cli は Redis のネイティブコマンドラインインターフェイスです。ApsaraDB Redis により、redis-cli を使用して既存の Redis データベースから ApsaraDB for Redis へのシームレスな移行が可能です。DTS を使用してデータをインポートすることも可能です。

注

- ・ ApsaraDB Redis は Alibaba Cloud イン트라ネットからのアクセスのみをサポートするため、Alibaba Cloud ECS インスタンスに対してのみ次の手順を実行可能です。Redis インスタンスが Alibaba Cloud ECS インスタンスにない場合は、データをインポートする前に既存の AOF を Alibaba Cloud ECS インスタンスにコピーします。
- ・ Redis-cli は Redis のネイティブコマンドラインインターフェイスです。ECS インスタンス上で redis-cli を使用できない場合は、redis-cli を使用する前に Redis をダウンロードしてインストールします。

手順

Alibaba Cloud ECS インスタンスに Redis インスタンスを作成済みの場合は、次の手順を実行します。

1. 既存の Redis インスタンス上で AOF 機能を有効にします (AOF 機能が有効になっている場合は、この手順をスキップします)。

```
# redis-cli -h old_instance_ip -p old_instance_port config set appendonly yes
```

2. AOF を使用して ApsaraDB Redis インスタンスにデータをインポートします (生成した AOF の名前を append.aof とします)。

```
# redis - cli - h aliyun_red is_instanc e_ip - p 6379 - a password -- pipe < appendonly . aof
```



:

ソース Redis インスタンスの AOF 機能を有効にしておく必要がない場合は、データのインポート後に次のコマンドを実行して機能を無効にします。

```
# redis-cli -h old_instance_ip -p old_instance_port config set appendonly no
```

5.2 ApsaraDB Redis からオンプレミス Redis への移行

5.2.1 バックアップセットの移行

redis-port ツールを使用して、ApsaraDB for Redis バックアップセットをオンプレミスの Redis データベースに移行します。

ApsaraDB Redis コンソールからバックアップセットデータのダウンロード

1. [ApsaraDB Redis コンソール](#)にログインし、インスタンス一覧ページでターゲットインスタンスを見つけます。
2. インスタンス ID をクリックするか、"操作" の縦のドットをクリックして、ショートカットメニューから [管理] を選択し、インスタンス情報ページに移動します。
3. アーキテクチャ図で DB ノード数を表示します。
4. バックアップとリカバリページで、DB ノード数に基づいてバックアップセットデータをダウンロードします。

redis-port のダウンロード

Redis-port アドレス

例

```
./ redis - port restore -- input = x / dump . rdb -- target
= dst \ _host : dst \ _port -- auth = dst \ _password [--
filterkey =" str1 | str2 | str3 " ] [-- targetdb = DB ] [-- rewrite ]
[-- bigkeysizes = SIZE ] [-- logfile = REDISPORT . LOG ]
```



注:

各データベースのバックアップセットに対してリカバリ手順を一度実行する必要があります。

パラメーター説明

- ・ x/dump.rdb: ApsaraDB for Redis バックアップセットのダンプファイルパスを示します。
- ・ dst_host: ユーザーが作成した Redis データベースのドメイン名または IP アドレスを示します。
- ・ dst_port: ユーザーが作成した Redis データベースのポートを示します。
- ・ dst_password: ユーザーが作成した Redis データベースのパスワードを示します。
- ・ str1|str2|str3: キーを str1、str2、または str3 でフィルターします。
- ・ DB: ユーザーが作成した Redis データベースと同期するデータベースを示します。
- ・ rewrite: すで書き込まれているキーを上書きします。
- ・ bigkeysizes=SIZE: 書き込まれた値が SIZE よりも大きい場合、ビッグキー書き込みモードが使用されていることを示します。

redis-port ログに基づくデータリカバリステータスの確認

```
./bin/redis-port restore --input=... --target=... --auth=...
2018/01/24 10:31:20 [INFO] set ncpu = 24, parallel = 24 filterdb = 0 targetdb = -1
2018/01/24 10:31:20 [INFO] restore from '/home/pan.liangp/run_4.0/dump_test.rdb' to '10.218.144.110:8179'
2018/01/24 10:31:20 [INFO] Aux information key:redis-ver value:4.0.2
2018/01/24 10:31:20 [INFO] Aux information key:redis-bits value:64
2018/01/24 10:31:20 [INFO] Aux information key:ctime value:1516359975
2018/01/24 10:31:20 [INFO] Aux information key:used-mem value:440742200
2018/01/24 10:31:20 [INFO] Aux information key:repl-stream-db value:0
2018/01/24 10:31:20 [INFO] Aux information key:repl-id value:e19e45132deb3be68a1e97a17fced7108fd1e02
2018/01/24 10:31:20 [INFO] Aux information key:repl-offset value:46308728805
2018/01/24 10:31:20 [INFO] Aux information key:aof-preamble value:0
2018/01/24 10:31:20 [INFO] db_size:2977 expire_size:0
2018/01/24 10:31:21 [INFO] total = 177007551 - 40387120 [ 22%] entry=770
2018/01/24 10:31:22 [INFO] total = 177007551 - 80511868 [ 45%] entry=863
2018/01/24 10:31:23 [INFO] total = 177007551 - 126672277 [ 71%] entry=1472
2018/01/24 10:31:24 [INFO] total = 177007551 - 159105721 [ 89%] entry=1472
2018/01/24 10:31:25 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:26 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] restore: rdb done
```

restore : rdb done が表示されたら、データリカバリは完了です。

5.3 ApsaraDB Redis インスタンス間の移行

5.3.1 redis-port ツールで異なるアカウントの ApsaraDB for Redis インスタンス間のデータの移行

redis-port ツールを使用して、異なる Alibaba Cloud アカウントにある ApsaraDB for Redis インスタンスからもう一方のインスタンスにデータを移行できます。

- ・ Linux ベースの ECS (Elastic Compute Service) インスタンスが、対象となる ApsaraDB for Redis インスタンスの属する VPC で作成されている必要があります。
- ・ 作成した ECS インスタンスに **redis-port** がダウンロードされている必要があります。
- ・ `chmod u+x redis-port` コマンドを実行して、redis-port を実行ファイルに変更する必要があります。
- ・ redis-port があるディレクトリで `mkdir logs` コマンドを実行する必要があります。

1. [ApsaraDB for Redis コンソール](#)にログインします。
2. インスタンス一覧 ページで、ソースの ApsaraDB for Redis インスタンスを検索します。インスタンス ID をクリックするか、"操作" の垂直に並んだ点をクリックして、ショートカットメニューから [管理] を選びます。
3. 左側のナビゲーションウィンドウから [バックアップと復元] をクリックします。
4. 対象となるバックアップファイルをバックアップファイルリストで検索して、"操作" から [ダウンロード] をクリックします。



注:

すぐにバックアップファイルを作成するには、バックアップと復元ページの右上隅の [バックアップの作成] をクリックします。表示された [インスタンスのバックアップ] メッセージボックスで、[確認] をクリックします。

5. 表示された [バックアップファイルのダウンロード] ダイアログボックスで、[イントラネット用 URL の取得] をクリックします。
6. ECS インスタンスで、前の手順でコピーしたアドレスからバックアップファイルをダウンロードします。



注:

クラスタータイプの ApsaraDB for Redis インスタンスでは、サブノードの数を基に複数のバックアップファイルが生成され、生成されたすべてのファイルをダウンロードする必要があります。

7. 以下のコマンドを実行して、すべてのバックアップを対象となるデータベースにインポートします。

```
./ redis - port restore - i backup file name - t domain  
name or IP address of the target database : port  
number -- auth = ' password of the target database '
```

"restore: rdb done" が表示された場合、データのインポートは成功しています。移行が完了しました。

5.3.2 Rump を使用した ApsaraDB Redis データベース間のデータ移行

Rump を使用して、同じ ApsaraDB Redis インスタンス内のデータベース間、または異なる ApsaraDB Redis インスタンス間でデータを移行できます。

- ・ 対象の ApsaraDB for Redis インスタンスが存在する VPC に Linux ベースの ECS (Elastic Compute Service) インスタンスが作成されている必要があります。
- ・ 作成した ECS インスタンス内に **Rump** がダウンロードされている必要があります。Rump のファイルタイプを実行可能ファイルに変更済みです。

クラウドサービスプロバイダーから Redis データを移行すると、以下の問題が発生します。

- ・ `slaveof` や `bgsave` などのデータ取得コマンドはサポートしていません。
- ・ `keys` コマンドは、サーバー上で輻輳を引き起こし、運用サービスに影響を与える可能性があります。

Rump の移行メカニズム

Rump は単一の `scan` コマンドを実行して、ソース ApsaraDB for Redis インスタンスから複数のキーリストを取得します。Rump は `dump` コマンドでキーコンテンツを取得し、`pttl` コマンドによってキーの有効期限を取得します。次に、Rump は `restore` コマンドを使用して、パイプラインモードでターゲット ApsaraDB for Redis インスタンスにキーを同期します。

利点

- ・ `keys` コマンドを `scan` コマンドに置き換えて、サーバー上での輻輳を防止します。
- ・ どのタイプのデータでも同期します。
- ・ 一時ファイルは使用しません。
- ・ バッファを持つチャネルを使用して、ソースサーバーのパフォーマンスを向上させます。
- ・ パイプラインモードを採用し、ネットワーク帯域幅を節約します。

ECS インスタンスで次のコマンドを実行してデータを同期します。

```
rump - from source_addr r - fromPwd source_pwd - to
dest_addr - toPwd dest_pwd [- size size ] [- replace ]
```

表 5-3: パラメーターの説明

パラメーター	説明
source_addr	redis :// host : port / db の形式のソース ApsaraDB for Redis インスタンスのアドレス。host と port は必須パラメーターで、db はオプションパラメーターです。db を指定していない場合は、デフォルトで値 0 が使用されます。
source_pwd	ソース ApsaraDB for Redis インスタンスのパスワード。
dest_addr	source_addr と同じ形式のターゲット ApsaraDB for Redis インスタンスのアドレス。
dest_pwd	ターゲット ApsaraDB for Redis インスタンスのパスワード。
size	一度に同期するキー数。デフォルト値: 10。
-replace	既存のキーが新しいキーと同じ場合、既存のキーを上書きするかどうかを示します。このパラメーターを指定すると、既存のキーが上書きされます。このパラメーターを指定していない場合、既存のキーは上書きされず、エラーメッセージが表示されます。

- ・ 例 1: 同じ ApsaraDB for Redis インスタンス内の db0 から db1 にデータをインポート

```
rump - from redis :// r - 123456789 . redis . rds .
aliyuncs . com : 6379 / 0 - fromPwd from_pass - to
redis :// r - 123456789 . redis . rds . aliyuncs . com :
6379 / 1 - toPwd to_pass - size 100
```

- ・ 例 2: Redis インスタンス A の ApsaraDB の db0 から Redis インスタンス B の ApsaraDB の db1 にデータをインポート

```
rump - from redis :// r - 123456789 . redis . rds .
aliyuncs . com : 6379 / 0 - fromPwd from_pass - to
redis :// r - 999999999 . redis . rds . aliyuncs . com :
6379 / 1 - toPwd to_pass - size 100
```

6 アラーム設定

背景

ApsaraDB for Redis では、インスタンスモニタリング機能が提供され、インスタンスで例外が検出されると SMS メッセージが送信されます。

モニタリングとアラームは Cloud Monitor を介して実装されています。Cloud Monitor ではメトリクスが設定され、メトリクスのアラームポリシーがトリガーされると、アラーム送信先グループ内のすべての送信先に通知されます。アラームが発生したら関連する送信先に迅速に通知されるように、アラームメトリックに対応するアラーム送信先を管理します。

手順

1. [Redis コンソール](#) をクリックし、対象のインスタンスを検索します。
2. インスタンス ID または [管理] をクリックし、[インスタンス情報] ページに移動します。
3. 左側のナビゲーションウィンドウで [アラーム設定] を選択します。
4. [アラーム設定] をクリックし、Cloud Monitor コンソールに移動します。[更新] をクリックし、モニタリングメトリクスの現在のステータスを手動で更新します。
5. [アラームルール] > [アラームルールの作成] の順に選択します。
6. [バッチアラームルール設定] ページでアラームルールを追加します。
7. [次へ] をクリックし、通知オブジェクトを設定します。[連絡先グループをすぐに作成する] をクリックし、アラーム連絡先またはアラーム連絡先グループを作成します。
8. [確認] をクリックし、[閉じる] をクリックします。



アラームルールの設定が完了したら、Cloud Monitor コンソールのアラームルールページでアラームルールを変更、無効化、および削除できます。アラームルールページでは、アラーム履歴が表示されます。