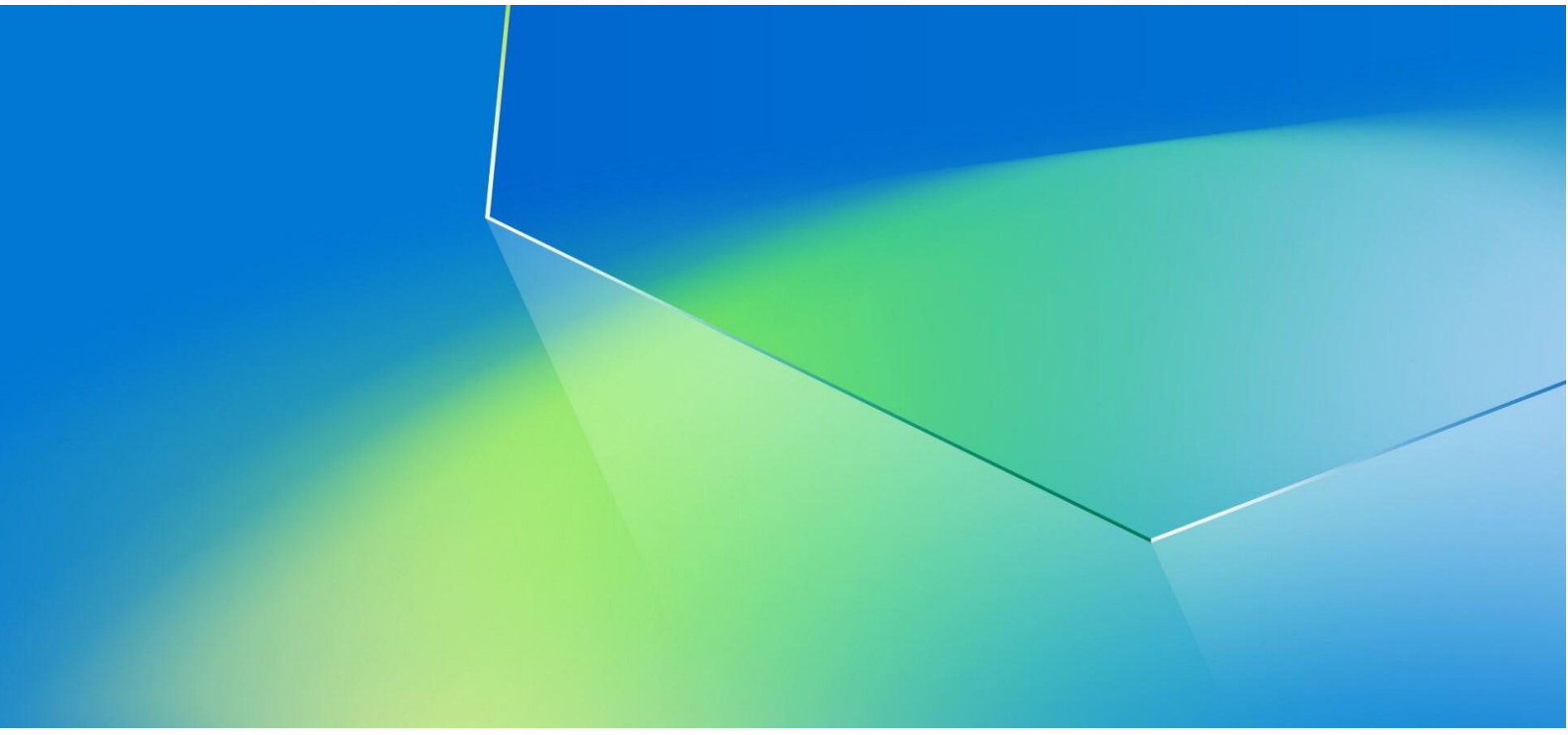
Microsoft Azure operated by 21Vianet

Azure In China

User Migration Guide

V1.0



Published in April 2025

Table of Contents

Migration Guide for Azure In China 3
Region Selection User Guide 4
Migration Process 6
Azure Resource Cross-Region Migration Guide 8
Migrate Azure API Management 11
Migrate Azure App Service 12
Migrate Azure Functions 19
Migrate Azure Virtual Machine 21
Migrate Azure Virtual Machine Scale Sets (VMSS) 24
Migrate Azure Batch services 26
Migrate Azure Cloud Services 30
Migrate Azure Application Gateway 34
Migrate Azure ExpressRoute 36
Migrate Azure Load Balancer 37
Migrate Azure Network Watcher 47
Migrate Azure Virtual Network 48
Migrate Virtual WAN 53
Migrate Azure NAT Gateway 60
Migrate Azure VPN Gateway 61
Migrate Web Application Firewall (WAF) 63
Migrate Azure Bastion Resource 64
Migrate Azure Container Registry 66
Migrate Azure Service Fabric 69
Migrate Azure Analysis Services 73
Migrate Azure Backup Resource 74
Migrate Azure Cache for Redis Instance 77
Migrate Azure Database for MySQL 81
Migrate Azure Database For PostgreSQL 82
Migrate Azure SQL Resource 83
Migrate SQL Server Stretch Database 85
Migrate Azure Monitor 86
Migrate Azure Automation 88
Migrate Azure Firewall 89
Migrate Azure Key Vault 94
Migrate Azure Managed Disks 99
Migrate Azure Storage Account 102
Migrate Azure Event Hubs 106

Migrate Azure Notification Hubs 108
Migrate Azure Service Bus 111
Migrate Azure IoT Hub 113
Migrate Azure Stream Analytics Jobs 118
Migrate Azure Logic Apps 124
Migrate Media Services 126
Migrate Azure Site Recovery 127

Migration Guide for Azure In China

Microsoft Azure operated by 21Vianet (referred to as Azure in China) is a public cloud platform independently operated in mainland China by Shanghai Blue Cloud Technology Co., Ltd. ("21Vianet"), a wholly owned subsidiary of Beijing 21Vianet Broadband Data Center Co., Ltd. It is physically and logically independent from Azure services operated by Microsoft in other regions around the world. Using Azure technology that serves globally, it provides consistent service quality assurance to customers. Currently, data centers are deployed individually in the eastern and northern parts.

Theoretically, cloud computing resources can be approximately considered infinite under certain conditions. However, due to the rapid development of cloud computing technology and the high growth rate of China's internet business, the construction speed and mode of cloud service infrastructure need continuous dynamic adjustments based on demand to meet the ever-growing and changing business needs. Users need to fully understand the characteristics of cloud services and adapt and adjust the systems built on cloud services to improve the flexibility and stability of the systems, addressing potential risks brought about by such dynamic needs.

This guide is designed to help Azure in China users securely and efficiently migrate existing Azure resources from North 1 and East 1 regions to other regions in China. It includes the following contents:

- Region Selection Guide
- Detailed Migration Process
- Resource Migration Manual

Region Selection User Guide

Regional Characteristic Differences

Due to significant differences in customer regional distribution, industry affiliation, product/service/IT system cycles, and other dimensions, each region of Azure in China evolves unique characteristics in response to customer needs. Understanding these differences and leveraging them during the system implementation, expansion, upgrade, and reconstruction processes can help architect systems that adapt long-term to the deployment region and reduce potential impacts caused by external factors.

Regional Service Differences

The deployment progress of specific services in each region of Azure in China varies based on customer characteristics and demand. Detailed differences can be referenced from Azure China Services Availability, which is regularly updated to ensure users have the latest information. Additionally, visit the Azure Updates page to get all announcements on new Azure services.

Industry Distribution Differences

Some industry customers' business or users may concentrate in certain hotspot regions or during specific periods, causing a concentrated release of resource demand in these areas or periods, leading to resource tension.

- **Retail Industry**: The retail industry is a key sector for China's cloud business. With the rapid growth of the Chinese consumer market and the booming development of new retail-related businesses over recent years, both local and overseas companies develop China as an important global consumer market. Especially in the Yangtze River Delta in eastern China and the Pearl River Delta in the south, retail businesses are focused areas of development. Therefore, cloud resources deployed in China's southeastern regions see significant consumption by retail customers, particularly during peak retail activity seasons like "Double 11", causing intense resource demand during these periods.

- **Advanced Resource Distribution Differences**: For advanced resources like SSD disks and GPU servers, due to the high initial investment, their deployment speed is based on analyzing potential business demand in each region and then phased investment according to growth trends. Hence, compared to ordinary resources, advanced resource prices are relatively higher, and quantities are limited. However, reasonable use of advanced resources can effectively reduce operational costs and improve system robustness.

    - Premium Managed Disks: Premium managed disks are based on high-performance SSD storage hardware. If heavily used in some regions, it can create temporary resource tension, impacting business development of customers in that region.
    - GPU Resources: GPU resources are now available in North 2, North 3, and East 2 regions. If users need GPU resources, it is recommended to use the North 3 region (for supported GPU models, refer to: NC Series and NV Series SKU List, need to request quota in destination regions).
    - Multi-AZ Deployment: The North 3 region in China offers Availability Zone high availability capabilities, providing 99.99% high availability assurance for critical business applications and better disaster recovery abilities.

Optimizing Regional Resources

You can adjust your deployment based on the following principles to reduce or eliminate risks.

Efficiently Utilizing Hotspot Regions

For Azure in China users whose customers are not primarily in the eastern regions and have low latency requirements, it is advisable to use resources in the two northern regions to mitigate potential business impacts due to resource tension caused by hotspot industry's peak activities.

Efficient Usage of Hotspot Resources

For advanced resources, use them wisely based on your business-specific metrics. Unless ordinary resources cannot meet your business requirements, avoid unnecessary use of advanced resources to enhance system robustness and reduce operating costs.

For example:

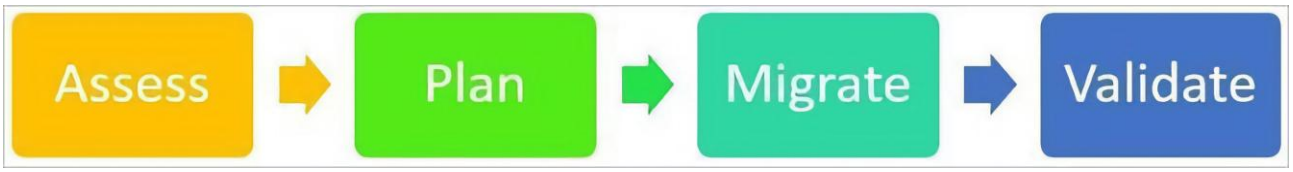
- If a Standard HDD can meet the requirements, choose **Standard HDD** disk for **OS disk type** when creating VM resources instead of the default **Premium SSD** option to reduce reliance on premium managed disks.
- Choose **Standard HDD** for **account type** when creating snapshots instead of **Premium SSD**.

For more information:

- Refer to the Managed Disks Pricing Details to learn more about the metrics and pricing differences between Standard HDD and Premium SSD.

Migration Process

This chapter provides guidelines to help you understand the overall process of migrating Azure resources from one Azure region to another. The focus of the migration process is on how to formulate an overall migration plan for your applications and then sequentially migrate Azure resources according to the migration plan. The steps in the migration process are as follows:



Migration Process

For more information, see Cloud Adoption Framework - Relocate Cloud Workloads.

Assessment

You should gather Azure account owners, subscription administrators, tenant administrators, and finance and accounting teams to understand the scale of the organization planning the migration. This is very important. Staff in these roles can provide comprehensive information on Azure usage in large organizations. During the assessment phase, compile a list of resources:

- Each subscription administrator and tenant administrator should run a series of scripts to list resource groups, the resources in each resource group, and the deployment settings of the resource groups in the environment.
- You should document the dependencies between applications in Azure and external systems.
- You should also document the number and amount of data for each Azure resource associated with each instance you plan to migrate.
- Ensure that the application architecture documentation is consistent with the Azure resource list.

By the end of this phase, you will have:

- A complete list of Azure resources to migrate.
- A list of dependencies between resources.
- Information on the complexity of the migration effort.

Planning

In the planning phase, you should complete the following tasks:

- Use the output of the dependency analysis completed during the assessment phase to define related components. Consider migrating related components through migration packages.
- Determine the target environment in the target Azure region.
- Identify the target Azure tenant. If there are new management requirements for the migration target environment (such as isolation, security, etc.), you can create a new Azure tenant.
- Identify the target subscription. If there are new management requirements for the migration target environment (such as resource management, cost management, etc.), you can create a new subscription.
- Select the target Azure region.
- Perform test migration scenarios to match the architecture in the source Azure region with the architecture in the target region.
- Determine a suitable migration timeline and schedule. Create user acceptance test plans for each migration package.

Migration

During the migration phase, use the tools, techniques, and recommendations discussed in the following sections to migrate or create new resources in the target region. Then, configure the applications.

Verification

In the verification phase, complete the following tasks:

- Complete user acceptance testing.
- If applicable, sync the latest data to the target environment.
- Ensure the application works as expected.
- Switch to the new application instance in the target region.
- Verify that the production environment works as expected.
- De-allocate resources in the source region.

Terminology

The following sections use these terms:

Source describes where the resources are being migrated from

- **Source Tenant Name**: The name of the tenant in the source Azure region (everything after @ in the account name).
- **Source Tenant ID**: The ID of the tenant in the source Azure region. The tenant ID appears when you hover the mouse over the account name in the upper right corner of the Azure portal.
- **Source Subscription ID**: The ID of the resource subscription in the source Azure region. You can have multiple subscriptions in the same tenant. Always ensure you are using the correct subscription.
- **Source Region**: The region from which resources are being moved.

Destination or Target refers to the location receiving the migrated resources

- **Target Tenant Name**: The name of the tenant in the target Azure region.
- **Target Tenant ID**: The tenant ID in the target region.
- **Target Subscription ID**: The subscription ID of the resources in the target region.
- **Target Region**: The region receiving the migrated resources.

Note

- Verify that the Azure services you are migrating are available in the target region.
- China North 3 Region supports multiple availability zone services. For the specific list of supported services, please refer to: Availability zone service and regional support.
- Before migrating, ensure that you have conducted thorough testing and backups to prevent data loss.

Cross-Region Migration of Azure Resources

For information on how to migrate specific resources across regions, refer to the Azure Resource Cross-Region Migration Manual chapter.

Azure Resource Cross-Region Migration Guide

This chapter describes the migration steps related to each migration phase.

Resource Assessment

- Understand the consistency of services and configurations between the source environment and the target environment.
- Estimate the operational cost in the target environment.
- Understand the dependencies between already deployed resources.
- Develop a comprehensive assessment report for decision-making and work planning.

Migration Plan

- Plan the migration order of resources based on the dependencies between resources.
- Estimate the time required for migration according to the migration order of resources.
- Generate a migration plan report.

Resource Migration Methods

Resource migration must follow the user-confirmed migration plan, migrating the deployed resources one by one in the specified order.

Use Azure Resource Mover

Using Resource Mover, you can currently move the following resources across regions:

- Azure VMs and associated disks (Azure Spot VMs are not currently supported)
- Encrypted Azure VMs and associated disks. This includes VMs with Azure disk encryption enabled and Azure VMs using default server-side encryption (both with platform-managed keys and customer-managed keys)
- NICs
- Availability sets
- Azure virtual networks
- Public IP addresses (Public IP will not be retained across regions)
- Network security groups (NSGs)
- Internal and public load balancers
- Azure SQL databases and elastic pools

For more information, see Azure Resource Mover.

Manual Migration

- Migrate security resources
  - Azure Firewall
  - Key Vault
- Migrate storage resources
  - Managed Disks
  - Storage Accounts
- Migrate analytic services
  - Azure Stream Analytics
  - Azure Synapse Analytics

  - HDInsight
- Migrate management tools
  - Azure Automation
  - Azure Monitor
  - Azure Site Recovery
  - Azure Backup
- Migrate integration resources
  - Event Hubs
  - Event Grid
    - Event Grid Domain Migration
    - Event Grid System Topics Migration
    - Event Grid Custom Topics Migration
  - Logic Apps
  - Notification Hubs
  - Service Bus
- Migrate compute resources
  - Azure Functions
  - Azure Virtual Desktop
  - Batch
  - Cloud Services
  - Virtual Machines
  - Virtual Machine Scale Sets
- Migrate network resources
  - Application Gateway
  - Azure Bastion
  - Azure DNS
  - Azure ExpressRoute
  - Azure Public IP
  - Azure Route Server
  - Azure Traffic Manager
  - Azure WAF
  - Load Balancer
  - NAT Gateway
  - Network Watcher
  - Private Link Service
  - Virtual Network
  - Virtual WAN
  - VPN Gateway
- Migrate media resources
  - Media Services
- Migrate container resources
  - Azure Container Registry
  - Azure Service Fabric
- Migrate database resources
  - Azure Analysis Services
  - Azure Cache for Redis
  - Azure Database for MySQL

  - Azure Database for PostgreSQL
  - Azure SQL
  - SQL Server Stretch Database
- Migrate web resources
  - API Management
  - App Service
- Migrate IoT resources
  - Azure IoT Hub
  - Notification Hubs

Migrate Azure API Management

To move an API Management instance from one Azure region to another, use the service's backup and restore operations. You can use a different API Management instance name or the existing name.

Considerations

- Select the same API Management pricing tier for both the source and target regions.

Prerequisites

- Review the requirements and limitations of the API Management backup and restore operations.
- Review what is not backed up. Note the settings and data that need to be manually recreated after moving the instance.
- Create a storage account in the source region. You will use this account to back up the source instance.

Prepare and Move

Option 1: Using Another API Management Instance Name

1. Create a new API Management instance in the target region using the same pricing tier as the source API Management instance. Use a different name for the new instance.
2. Back up the existing API Management instance to the storage account.
3. Restore the backup of the source instance to the new API Management instance.
4. If you have a custom domain pointing to the source region API Management instance, change the custom domain CNAME to point to the new API Management instance.
5. If you have content that is not backed up, please recreate it manually.

Option 2: Using the Same API Management Instance Name

Note: This option will delete the original API Management instance and cause downtime during migration. Ensure you have a valid backup before deleting the source instance.

1. Back up the existing API Management instance to the storage account.
2. Delete the API Management instance in the source region.
3. Create a new API Management instance in the target region using the same name as the source region.
4. Restore the backup of the source instance to the new API Management instance in the target region.
5. If you have content that is not backed up, please recreate it manually.

Validate

1. Ensure the restore operation is successfully completed before accessing the API Management instance in the target region.
2. Configure the settings that are not automatically moved during the restore operation. Examples: virtual network configuration, managed identities, developer portal content, and custom domains and custom CA certificates.
3. Access the API Management endpoint in the target region. For example, test an API or access the developer portal.

Migrate Azure App Service

Table of Contents

Azure App Service Regional Migration Steps

- Prerequisites
- Migration Preparation
- Migration Plan
- Relocation
- Validation
- Cleanup

Azure Web App Regional Migration

- Overview
- Pre-Migration Preparation
- Step 1: Backup Existing Web App Service
- Step 2: Create Resources in Target Region
- Step 3: Deploy App Service to New Region
- Step 4: Validate Deployment
- Step 5: Switch Traffic and Monitor
- Step 6: Cleanup Old Resources
- References

Azure App Service Regional Migration Steps

Azure App Service enables you to build and host web app, mobile backends, and RESTful APIs in the programming language of your choice without managing infrastructure. It provides autoscaling, high availability, and supports both Windows and Linux, with automated deployment from GitHub, Azure DevOps, or any Git repository.

App service resources are region-specific and cannot be moved across regions. You must create a copy of your existing app service resources in the target region and then relocate the content to the new application. If your source application uses a custom domain, it can be migrated to the new application in the target region after relocation is completed.

Prerequisites

- Ensure the target region supports your application service and any related services.
- Verify that you have sufficient permissions to deploy application service resources to the target subscription and region.
- Check for any regional restrictions assigned by Azure policies.
- Consider operational costs as compute resource pricing may vary by region. To estimate potential costs, refer to the Pricing Calculator.

Migration Preparation

Identify all app service resources you are currently using. For example:

- App Service Environment
- App Service Plan

- Deployment slots
- Custom domains purchased in Azure
- TLS/SSL certificates
- Azure Virtual Network Integration
- Hybrid Connections
- Managed identities
- Backup settings

Some resources (for example, imported certificates or hybrid connections) include integration with other Azure services. Refer to the documentation of the respective service for information on how to move these resources across regions.

Migration Plan

Here are the planning checklists and considerations for the migration:

- State, storage, and downstream dependencies
- Certificates
- Configuration
- VNet connection/custom names/DNS
- Identity
- Service endpoints

State, Storage, and Downstream Dependencies

- Determine whether your application service app is stateful or stateless.
- Check for internal caching and state in the application code.
- Disable session affinity settings. Where possible, it is recommended to disable session affinity settings. Disabling session affinity can improve load balancing for horizontal scaling. Any internal state may impact direct workload transition planning, especially in zero-downtime requirements. Where feasible, refactor any application state to make the application stateless in preparation for the move.
- Analyze database connection strings. Database connection strings can be found in the application settings, but they can also be hard-coded or managed within configuration files included with the application. For higher-level workload transition planning, analyze and plan data migration/replication. For chatty or latency-sensitive applications, having the application in the target region return to data sources in the source region will perform poorly.
- Analyze external caches (e.g., Redis). Application caches should be as close to the application deployment as possible. Analyze how the cache is populated, expiration/eviction policies, and the impact on the first user access to the workload following the direct transition.
- Analyze and plan API (or application) dependencies.
- Analyze and plan regional services. Application Insights and Log Analytics data are regional. Consider creating new Application Insights and Log Analytics storage in the target region. For App Insights, having new resources also means that connection strings must be updated as part of application configuration changes.

Certificates

Application service certificate resources can be moved to a new resource group or subscription but cannot be moved across regions. Certificates that can be exported can be imported into the app or a Key Vault in the new region. This export-and-import process is equivalent to moving between regions.

Configuration

- Take a snapshot of existing application settings and connection strings from the Azure portal. Expand “Settings” > “Environment Variables,” select “Advanced edit” under “Application Settings” or “Connection Strings,” and save the JSON output containing the existing settings or connections. These settings need to be recreated in the new region, but the values themselves may change due to subsequent region changes in connection services.

- Existing Key Vault references cannot be exported across Azure geo boundaries. Any necessary references must be recreated in the new region.

- Application configuration can be managed via Azure App Configuration or another central (downstream) database dependency. Review any application configuration store or similar store for environment-specific and region-specific settings that may need modification.

- Ensure to check any disk file configurations, which may or may not be overridden by application settings.

VNet Connection/Custom Names/DNS

- Application Service Environment is a VNet-injected single-tenant service. App Service Environment networking differs from multi-tenant App Service, which requires one or two “private endpoints” or “regional VNet integration”. Other options that may be in play include legacy P2S VPN-based VNet integration and Hybrid Connections (Azure Relay Service).

- Recreate private endpoints in the target region (if used). This also applies to regional VNet integration.

- DNS for App Service Environment is typically managed via customer-specific custom DNS solutions (each app has a manual setting override). App Service Environment provides load balancers for ingress/egress, while App Service itself filters by host header. As a result, multiple custom names can point to the same App Service Environment ingress endpoint. App Service Environment does not require domain verification.

Identity

- System-assigned managed identities need to be recreated in the new target region along with the application. Typically, Microsoft Entra ID applications auto-created (used by EasyAuth) default to the application resource name.

- User-assigned managed identities cannot be moved across regions either. To retain user-assigned managed identities in the same resource group as the application, they must be recreated in the new region.

- For relocated services, grant managed identities the same permissions as the original identities being replaced, including group memberships.

- Plan to relocate your identity provider (IDP) to the target region. While Microsoft Entra ID is a global service, some solutions rely on local (or downstream) IDP.

- Update any resources that may rely on Kudu FTP credentials for application service.

Service Endpoints

VNet service endpoints for Azure Application Service restrict access to a specified virtual network. Additionally, these endpoints can restrict access to a range of IPv4 (Internet Protocol version 4) addresses. Any user attempting to connect from outside of those ranges to Event Hubs won’t be able to access these resources. If service endpoints are configured on Event Hubs resources in the source region, they need to be likewise configured in the target region.

To successfully recreate Azure Application Service in the target region, the VNet and subnets must be pre-created. Service endpoints won’t automatically configure when moving all these resources via the Azure

Resource Mover tool. Therefore, manually configure service endpoints, which can be done through the Azure portal, Azure CLI, or Azure PowerShell.

Relocation

To relocate application service resources, you can use the Azure portal or Infrastructure as Code (IaC). During the migration process, the source application can be accessed normally. After verifying that the application in the target region is functioning correctly, the traffic needs to be redirected to the target region. This process may result in the application being temporarily unavailable to external users.

Relocation using the Azure Portal

The greatest benefit of using the Azure portal for relocation is its simplicity. The application, plan, content, and many settings will be cloned to a new application service resource and plan.

To relocate application service resources to a new region using the Azure portal:

1. Create a backup of the source application.
2. Create an application in a new application service plan in the target region.
3. Restore the backup in the target application.
4. If you are using a custom domain, pre-bind it to the target application using `asuid.` and enable the domain in the target application.
5. Configure everything else in the target application to match the source application and validate your configuration.
6. When ready to point the custom domain to the target application, remap the domain name.

Relocation using IaC

Use IaC when CI/CD pipelines are existing or can be created. With CI/CD pipelines, application service resources can be created in the target region via Deployment Actions or Kudu zip deployment.

Validation

After relocation, test and validate Azure Application Service using recommended guidelines:

- Run smoke tests and integration tests after relocating Azure Application Service to the target region. Tests can be performed manually or run via script. Ensure to verify all configurations and dependent resources are correctly linked and that all configured data is accessible.
- Validate all Azure Application Service components and integrations.
- Conduct integration tests of the deployment in the target region, including all formal regression tests. Integration tests should align with the regular cadence of workload business deployment and testing processes.
- In some cases, especially if the relocation involves updates, changes to the application or Azure resources, or usage profile changes, use load testing to verify that new workloads meet their purpose. Load testing is also an opportunity to validate operational and monitoring coverage. For example, use load testing to verify the desired infrastructure and application logs are correctly generated. Load tests should be measured against an established workload performance baseline.

Cleanup

Delete the source application and its application service plan. Application service plans in non-free tiers incur charges, even if no applications are running within them.

Azure Web App Regional Migration

Overview

This manual aims to guide how to migrate an Azure Web App from one region to another. This process includes backing up existing application services, creating new resources in the target region, deploying the application service, validating the migration results, and more.

Pre-Migration Preparation

Before beginning the migration, ensure the following preparations are completed:

- Confirm that the target region supports Web App.
- Backup all related data and configurations.
- Ensure sufficient permissions for resource creation and management during migration.
- Understand the dependencies and configurations of the existing Web App.

Step 1: Backup Existing Web App

1. Sign in to the Azure Portal.
2. Navigate to the Web App you need to migrate.
3. In the left navigation bar under “Settings,” select “Backup.”
4. Configure the backup storage account and container and perform the backup operation.
5. Download the backup file and securely store it.

Azure CLI Example:

```
# Set variables
$RESOURCE_GROUP="your-resource-group"
$APP_NAME="your-api-app-name"
$BACKUP_NAME="your-backup-name"
$CONTAINER_URL="your-storage-account-url"
# Create backup
az webapp config backup create `
  --resource-group $RESOURCE_GROUP `
  --webapp-name $APP_NAME `
  --backup-name $BACKUP_NAME `
  --container-url $CONTAINER_URL
```

Step 2: Create Resources in Target Region

1. In the Azure portal, navigate to “Create a resource.”
2. Search for and select “Web App.”
3. Fill in the basic information to create the Web App and select the target region.
4. Configure the “App Service Plan,” ensuring you select the resource group in the target region.
5. Complete the creation and wait for the resource deployment to finish.

Azure CLI Example:

```
# Set variables
$TARGET_RESOURCE_GROUP="your-target-resource-group"
$TARGET_LOCATION="your-target-location"
$NEW_PLAN_NAME="your-new-app-service-plan"
```

```
$NEW_APP_NAME="your-new-api-app-name"
$SKU="your-new-sku"

# Create resource group (if it does not exist)
az group create `
  --name $TARGET_RESOURCE_GROUP `
  --location $TARGET_LOCATION

# Create new App Service plan
az appservice plan create `
  --name $NEW_PLAN_NAME `
  --resource-group $TARGET_RESOURCE_GROUP `
  --location $TARGET_LOCATION `
  --sku $SKU

# Create new Web App
az webapp create `
  --resource-group $TARGET_RESOURCE_GROUP `
  --plan $NEW_PLAN_NAME `
  --name $NEW_APP_NAME
```

Step 3: Deploy Application Service to New Region

1. Navigate to the newly created Web Application.
2. Under “Settings,” select “Backup.”
3. Configure the backup storage account and container, and restore the backup file to the new Web App.

Azure CLI Example:

```
# Set variables
$NEW_APP_NAME="your-new-api-app-name"
$NEW_RESOURCE_GROUP="your-new-resource-group"
$CONTAINER_URL="your-storage-account-url"
$BACKUP_NAME="your-backup-file-name"

# Restore backup to new Web App
az webapp config backup restore `
  --resource-group $NEW_RESOURCE_GROUP `
  --webapp-name $NEW_APP_NAME `
  --backup-name $BACKUP_NAME `
  --container-url $CONTAINER_URL `
  --overwrite
```

Step 4: Validate Deployment

1. In the Azure portal, navigate to the Web App in the new region.
2. Access the Web App’s URL to ensure the application is running correctly.
3. Check logs and monitoring metrics to ensure there are no errors or anomalies.

Azure CLI Example:

```
# Set variables
$NEW_APP_NAME="your-new-api-app-name"
$NEW_RESOURCE_GROUP="your-new-resource-group"

# Get the Web App URL
az webapp show `
  --name $NEW_APP_NAME `
  --resource-group $NEW_RESOURCE_GROUP `
  --query defaultHostName `
  --output tsv
```

Step 5: Switch Traffic and Monitor

1. Update DNS records or load balancer configurations to switch traffic to the Web App in the new region.
2. Monitor the application service post-traffic switch to ensure stable operation.
3. Use Azure Monitor and Application Insights for performance and health checks.

Step 6: Cleanup Old Resources

1. Confirm the Web App and deployed application service in the new region are running stably and without issues.
2. Navigate to the Web App in the source region.
3. Stop and delete the Web App service and related resources in the source region.
4. Delete backup files or storage accounts if they are no longer needed.

Azure CLI Example:

```
# Stop old Web App
az webapp stop `
  --name $APP_NAME `
  --resource-group $RESOURCE_GROUP

# Delete old Web App
az webapp delete `
  --name $APP_NAME `
  --resource-group $RESOURCE_GROUP

# Delete old App Service plan (if no other applications are using it)
az appservice plan delete `
  --name $PLAN_NAME `
  --resource-group $RESOURCE_GROUP
```

References

- Azure App Service Overview
- Azure App Service Documentation
- Azure Command Line Interface (CLI) Documentation

For any questions, please contact your Azure support team.

Migrate Azure Functions

Azure Functions resources are region-specific and cannot be moved across regions. You must create a copy of the existing function app resources in the target region and then redeploy the function code to the new app.

If minimal downtime is needed, consider running the function app in both regions to achieve a disaster recovery architecture:

- Azure Functions Geo-Disaster Recovery
- Disaster Recovery and Geo-Distribution in Azure Durable Functions

Prerequisites

- Ensure the target region supports Azure Functions and any related services whose resources you want to move
- Access to the original source code of the function to be migrated

Prepare

Identify all function app resources used on the source region, which may include the following:

- Function app
- Hosting plans
- Deployment slots
- TLS/SSL certificates and settings
- Network configuration options
- Managed identities
- Configured application settings — Users with sufficient access can use the advanced editing feature in the portal to copy all source application settings
- Scaling configurations

Functions can be connected to other resources using triggers or bindings. For information on how to move these resources across regions, refer to the respective service documentation.

You should also be able to export templates from existing resources.

Move

Deploy the function app to the target region and review the configured resources.

Redeploy Function App

If you have access to the deployment and automation resources that created the function app in the source region, rerun the same deployment steps in the target region to create and redeploy the app.

If you only have access to the source code but not the deployment and automation resources, you can deploy and configure the function app on the target region using any available deployment technology.

Review Configured Resources

If resources were not configured during deployment, review and configure the resources identified in the above Prepare steps in the target region.

Migration Considerations

- If the deployment resources and automation did not create the function app, create the same type of app in the target region in a new hosting plan.
- Function app names are globally unique in Azure, so the app in the target region cannot have the same name as the app in the source region.
- You need to check and update references and application settings that connect the function app to dependencies as needed. For example, when moving a database that the function calls, you must also update the application settings or configuration to connect to the database in the target region. Some application settings (such as Application Insights instrumentation keys or Azure Storage account used by the function app) might already be configured in the target region and may not need updating.
- Remember to validate the configuration and test the function in the target region.
- If custom domains are configured, remap the domain names.

Clean Up Source Resources

After the move is complete, delete the function app and hosting plan from the source region. Even if function apps in the Premium or Dedicated plans are not running, they still incur charges.

For More Information

See Move Function Apps Between Regions in Azure Functions

Migrate Azure Virtual Machine

Table of Contents

- Overview
- Migrating Virtual Machines Using Azure Resource Mover
  - Prerequisites
  - Downtime Impact
  - Migration Process
- Migrating Virtual Machines Using Azure Site Recovery
- Summary

Overview

This tutorial shows you how to move Azure Virtual Machines (VMs) to a different Azure region.

You can choose to migrate by using Azure Resource Mover or using Azure Site Recovery.

Migrating Virtual Machines Using Azure Resource Mover

To move Azure VMs to another region, we recommend using Azure Resource Mover.
Resource Mover provides:

- A single hub for moving resources across regions.
- Reduced move time and complexity. Everything you need is in a single location.
- A simple and consistent experience for moving different types of Azure resources.
- An easy way to identify dependencies across resources you want to move. This helps you to move related resources together, so that everything works as expected in the target region, after the move.
- Automatic cleanup of resources in the source region, if you want to delete them after the move.
- Testing. You can try out a move, and then discard it if you don't want to do a full move.

First, confirm whether the VM to be migrated is an **Encrypted VM**, as the migration process will differ slightly depending on whether the VM is encrypted.

Encrypted VMs can be described as either: * VMs that have disks with Azure Disk Encryption enabled. For more information, see Create and encrypt a Windows virtual machine with the Azure portal.
* VMs that use customer-managed keys (CMKs) for encryption at rest, or server-side encryption. For more information, see Use the Azure portal to enable server-side encryption with customer-managed keys for managed disks.

Prerequisites

Before starting, ensure that you meet the following prerequisites:

1. **Resource Mover Support**
   Review the supported regions and other common questions.

2. **Subscription Permissions**
   Check that you have **Owner** access on the subscription containing the resources that you want to move.
   > **Why do I need Owner access?**
   > *The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a system-assigned managed identity, formerly known as Managed Service Identify (MSI) that's trusted by the subscription.*

> *To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs Owner permissions on the subscription. Learn more about Azure roles.*

3. **VM Support**
   - Check that the VMs you want to move are supported.
   - Verify supported Windows VMs.
   - Verify supported Linux VMs and kernel versions.
   - Check supported compute, storage, and networking settings.
4. **Destination subscription**
   The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have a quota, request additional limits.
5. **Destination region charges**
   Verify pricing and charges associated with the target region to which you're moving VMs. Use the pricing calculator to help you.

For **encrypted VMs**, additional prerequisites of the following is required:

1. **Key Vault Requirements (Azure Disk Encryption)**
   If you have Azure Disk Encryption enabled for VMs, you require a key vault in both the source and destination regions. For more information, see Create a key vault.
   For the key vaults in the source and destination regions, you require these permissions:
   - Key permissions: Key Management Operations (Get, List) and Cryptographic Operations (Decrypt and Encrypt)
   - Secret permissions: Secret management operations (Get, List, and Set)
   - Certificate (List and Get)
2. **Disk Encryption Set (Server-Side Encryption with CMK)**
   If you're using VMs with server-side encryption that uses a CMK, you require a disk encryption set in both the source and destination regions. For more information, see Create a disk encryption set.
   Moving between regions isn't supported if you're using a hardware security module (HSM keys) for customer-managed keys.

Downtime Impact

During Initiate the move phase, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).

The table summarizes what's impacted when you're moving across regions.

Behavior

Across Regions

Data

Resource data and metadata are moved. Metadata is stored temporarily to track status of resource dependencies and operations.

Resources

The source resource stays intact to ensure that apps continue to work, and can optionally be removed after the move. A resource is created in the target region.

Move Process

Multi-step process requiring manual intervention and monitoring.

Testing

Testing the move is important, since the apps should continue to work as expected in the target region, after the move.

Downtime

No data loss expected, but some downtime to move resources.

For more information on the move process, status, and impact of moving, see About the move process.

Migration Process

Depending on whether the source VM is an **encrypted VM**, you can follow the steps described in Move VMs across regions or Move encrypted Azure VMs across regions using Azure Resource Mover to complete the VM migration.

Migrating Virtual Machines Using Azure Site Recovery

Now all China Azure regions are same geographic cluster supported by Azure Site Recovery (ASR) (See geographic cluster).
You can also follow the steps described in Move VMs to another Azure region to complete the VM migration. However, we recommend you use Azure Resource Mover for VM migration.

Summary

Read the document carefully and thoroughly before starting the migration and check each step carefully to avoid data loss or service disruption.

Before migrating in a production environment, test and validate in a test environment.

If you encounter any issues, refer to Azure official documentation or contact Azure technical support.

For more reference documents on Azure VMs:

- Virtual Machines in Azure

Migrate Azure Virtual Machine Scale Sets (VMSS)

Introduction

To migrate virtual machine scale sets across Azure regions, export the Resource Manager template, adapt it to the new environment, and then redeploy to the target region.

Export only the base template and redeploy the template in the new environment. Individual virtual machine scale set instances should all be the same.

Before starting the redeployment, ensure dependencies on other resources are understood and migrated to the target region.

Important:
Change locations, Key Vault secrets, certificates, and other GUIDs to be consistent with the new region.

Prerequisites

Before you begin, ensure that you have the following prerequisites:

- If the source VM supports availability zones, then the target region must also support availability zones. To see which regions support availability zones, see Azure regions with availability zone support.
- The subscription in the destination region needs enough quota to create the resources. If you exceeded the quota, request an increase. For more information, see Azure subscription and service limits, quotas, and constraints.
- Consolidate all the associated extensions from source Virtual Machine Scale Set, as some need to be reconfigured after relocation.
- Confirm if the VM image is a part of VM image gallery. Gallery resources need to be replicated to the target region.
- Capture the list of resources that are being configured, such as capturing diagnostic logs. This is important with respect to prioritization and sequencing.
- If the source region VMSS relies on the following resources, ensure that they are available and deployed in the target region:
  - Log Analytics workspace
  - Diagnostic virtual machine scale set
  - Key Vault
  - Proximity placement group
  - Public IP address
  - Load balancer
  - Virtual network
- Ensure that you have a Network Contributor role or higher in order to configure and deploy a Load Balancer template in another region.
- Identify the networking layout of the solution in the source region, such as NSGs, Public IPs, VNet address spaces, and more.

Migration

The general steps for migrating a virtual machine scale set across regions are as follows: 1. Manually set the source Virtual Machine Scale Set instance count to 0. 2. Export the source virtual machine scale set template. 3. Edit the template content according to the new environment. 4. In the target region, use IAC (Infrastructure as Code) tools such as Azure Resource Manager templates, Azure CLI, or PowerShell to recreate the virtual machine scale set using the exported template. 5. Associate the dependent resources to the target virtual

machine scale set, such as the Log Analytics workspace in the Monitoring section. Also, configure all the extensions consolidated in the Prerequisites section.

For more details, see Relocate Azure virtual machines scale set to another region

Downtime Impact

During the migration process, consider the following downtime impacts:

- **Instance Shutdown:** You need to manually set the source VMSS instance count to 0 during migration, leading to instance shutdown and affecting application availability.
- **Dependency Resource Reconfiguration:** Reconfiguring dependency resources in the target region may cause brief service interruptions, depending on the complexity and number of resources.
- **Data Synchronization Delay:** If data migration is involved, it might lead to temporary data desynchronization, affecting real-time data access.

To minimize downtime impact, consider the following preparations: - **Backup Data:** Ensure that all critical data is backed up before migration. - **Choose Off-Peak Times:** Perform the migration during business off-peak hours to minimize user impact. - **Test Environment Verification:** Conduct a complete migration test in a test environment to verify all steps and configurations.

Summary

Before migrating in a production environment, test and verify it in a test environment.

If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

For more information: * Refresh your knowledge by completing the Virtual Machine Scale Sets tutorial. * Learn how to export Azure Resource Manager templates. * Refer to the Azure Resource Manager overview. * Get an overview of Virtual Machine Scale Sets. * Read the Azure regions overview. * Learn how to redeploy templates.

Migrate Azure Batch services

Table of Contents

- Introduction
- Prerequisites
- Preparation
- Migration Steps
  - Export the Template
  - Modify the Template
  - Create a Batch Account
  - Configure the Batch Account
  - Data Migration
  - Verification and Testing
  - Clean Up Resources
- Summary

Introduction

This manual is designed to guide you on how to migrate the Azure Batch Service from one region to another. We will detail the migration steps and considerations to ensure a smooth transition.

In certain situations, you might want to move an existing Azure Batch service from one region to another. For example, you might want to migrate to implement a disaster recovery plan. While you can't directly transfer the Azure Batch service from one region to another, you can use an Azure Resource Manager template (ARM template) to export the existing configuration of the Azure Batch service. Then, use the ARM template to create the Azure Batch service in another region. First, export the Azure Batch service configuration as a template file. Next, modify the template file parameters to match the target region. Then, deploy the modified template to the new region. Finally, recreate jobs, job schedules, tasks, etc., in the Azure Batch service.

Prerequisites

- Ensure that the services and features used by your Azure Batch service are supported in the new target region.
- It is recommended to move any Azure resources associated with the Batch account to the new target region. For example, follow the steps to move the associated Azure storage account to another region. If necessary, you can keep the resources in the original region, but performance is generally better if the Batch account and other Azure resources used by workloads are in the same region. This article assumes that storage accounts or any other regional Azure resources have been migrated to align with the Batch account.

Preparation

Before starting, ensure you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Azure CLI installed and configured.
4. Ensure that the Batch service has no ongoing tasks and that data has been backed up.

Migration Steps

Export the Template

Export an ARM template that contains the Batch account settings and information.

1. Log in to the Azure Portal.
2. Select “All resources,” and then select your Batch account.
3. Select “Settings” > “Export template.”
4. Select the “Include parameters” checkbox.
5. Select “Download” on the “Export template” page.
6. Locate the .zip file downloaded from the Azure Portal and extract it locally.

   This zip file contains two files that make up the template: *template.json* and *parameters.json*.

Modify the Template

Load and modify the template so that a new Azure Batch service can be created in the target region.

1. In the Azure Portal, select “Create a resource.”
2. In “Search services and marketplace,” type “Template deployment” and press **ENTER**.
3. Select “Template deployment (deploy using custom templates).”
4. Select “Create.”
5. Select “Build your own template in the editor.”
6. Select “Load file,” and then choose the “*template.json*” file downloaded in the previous section.
7. Replace “” in the code below with the name of the target Batch account.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "batchAccounts_mysourceaccount_name": {
      "defaultValue": "<target-batch-account>",
      "type": "String"
    }
  },
```

8. Create an Azure resource group and an Azure storage account in the target region.
9. Replace “” in the code below with the target subscription ID if different.

   Replace “” with the target resource group.

   Replace “” with the target storage account.

```
  "storageAccounts_mysourcestorageaccount_externalid": {
        "defaultValue": "/subscriptions/<subscription-id>/resourceGroups/<target-resource-group>/providers/Microsoft.Storage/storageAccounts/<target-storage-account>",
        "type": "String"
    }
},
```

10. Replace “” in the code below with the target region.

```
{
    "resources": [
        {
            "type": "Microsoft.Batch/batchAccounts",
            "apiVersion": "2021-01-01",
            "name": "[parameters('batchAccounts_mysourceaccount_name')]",
            "location": "<target-location>",
```

11. After making the modifications, choose “Save” under the “*template.json*” file.

Create a Batch Account

1. Enter or select property values:
   - **Subscription**: Select the Azure subscription.
   - **Resource Group**: Select the target region resource group.
   - **Location**: Choose the Azure region where you want to move the Batch account.
2. Select “Review + create,” and then “Create.”

Configure the Batch Account

Some features in the Batch account will not be exported to the template, so they must be recreated in the new Batch account. These include:

- Jobs (and tasks)
- Job schedules
- Certificates
- Applications

Ensure these features are configured in the new account as needed. You can refer to the source Batch account to see how these features are configured.

Data Migration

If there is data to migrate (e.g., blobs stored in Azure Storage), copy it to the new storage account.

Verification and Testing

1. Start the newly created Batch service and verify that the configuration is correct.

```
# List pools and jobs in the new Batch account to ensure they match the information in the old account
az batch pool list `
--account-endpoint <NewBatchAccountEndpoint> `
```

```
    --account-key <NewBatchAccountKey> `
    --account-name <NewBatchAccountName>
    # List jobs in the new Batch account
    az batch job list `
    --account-endpoint <NewBatchAccountEndpoint> `
    --account-key <NewBatchAccountKey> `
    --account-name <NewBatchAccountName>
```

2. Deploy test tasks to ensure the new Batch service can process jobs correctly.

Clean Up Resources

Once you have confirmed that the new Batch service is functioning correctly, you can delete the old Batch service and related resources in the source region.

Summary

By following the above steps, you have successfully migrated the Azure Batch service from one region to another. Be sure to carefully check each step during the migration process to avoid data loss or service interruption. If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

Learn how to migrate the Azure Batch service. Related reference documents:

- Azure Batch Azure Tutorial
- Create Azure Batch Account in Azure Portal
- Create Azure Batch Account with ARM Template

If you have any questions, please contact your Azure support team.

Migrate Azure Cloud Services

Table of Contents

- Introduction
- Prerequisites
- Preparations
- Migration Steps
  - Stop Cloud Services
  - Prepare Configuration and Package Files
  - Create Cloud Services in Target Region
  - Validation and Testing
  - Clean Up Resources
- Summary

Introduction

This article introduces the solution for migrating **Azure Cloud Services (Extended Support)** from one region to another. Currently, migrating Azure Cloud Services (Extended Support) from one Azure region to another is not supported. Therefore, the provided solution involves redeploying the Azure Cloud Services (Extended Support) in the target region using service definition *(.csdef)*, service configuration *(.cscfg)*, and service package *(.cspkg)* files, thus achieving migration of Azure Cloud Services (Extended Support) to the target Azure region.

Currently, **Azure Cloud Services (classic) is deprecated** and will be discontinued for all customers on August 31, 2024. See Azure Cloud Services (classic) Overview.

For migrating Azure Cloud Services (classic) to Azure Cloud Services (Extended Support), please refer to the relevant Migration Overview.

Prerequisites

1. **Virtual Network**: Deployment of Azure Cloud Services (Extended Support) must be within a virtual network. You can create a virtual network using the Azure portal, Azure PowerShell, Azure CLI, or Azure Resource Manager templates (ARM templates). The virtual network and subnet must be referenced in the NetworkConfiguration section of the configuration (.cscfg) file.

   For virtual networks in the same resource group as the cloud service, referencing the virtual network name in the service configuration (.cscfg) file is sufficient. If the virtual network and the cloud services (extended support) are in two different resource groups, specify the full Azure Resource Manager ID for the virtual network in the configuration (.cscfg) file.

- Virtual Network in the same resource group:

```
<VirtualNetworkSite name="<vnet-name>"/>
    <AddressAssignments>
      <InstanceAddress roleName="<role-name>">
       <Subnets>
```

```
        <Subnet name="<subnet-name>"/>
      </Subnets>
    </InstanceAddress>
  </AddressAssignments>
```

- Virtual Network in different resource groups:

```
<VirtualNetworkSite name="/subscriptions/<sub-id>/resourceGroups/<rg-name>/providers/Microsoft.Network/virtualNetworks/<vnet-name>"/>
  <AddressAssignments>
    <InstanceAddress roleName="<role-name>">
      <Subnets>
        <Subnet name="<subnet-name>"/>
      </Subnets>
    </InstanceAddress>
  </AddressAssignments>
```

2. **Access Control**: The subscription containing the network resources must have the “Network Contributor” or higher role for Azure Cloud Services (Extended Support).

3. **Key Vault**: Azure Key Vault can store certificates associated with Azure Cloud Services (Extended Support). Add certificates to the key vault and then reference the certificate thumbprint in the configuration (.cscfg) file for deployment. You must also enable the key vault access policy (in the portal) for **Azure VMs used for deployment**, so that the Azure Cloud Services (Extended Support) resources can retrieve certificates stored as secrets in the key vault. You can create a key vault in the Azure portal or using PowerShell. The key vault must be created in the same region and subscription as the cloud service.

Preparations

Before starting, make sure you have:

1. A valid Azure subscription.
2. Azure CLI enabled and logged in.
3. The migration account has permissions for both the source and target regions.
4. Ensure that your Azure Cloud Services (Extended Support) does not have any running tasks and all data is backed up.
5. Have the latest service definition *(.csdef)*, service configuration *(.cscfg)*, and service package *(.cspkg)* files ready.

Migration Steps

Stop Cloud Services

First, stop the Azure Cloud Services (Extended Support) resources you want to migrate.

```
az cloud-service power-off `

--resource-group <ResourceGroupName> `
```

```
--cloud-service-name <CloudServiceName>
```

Prepare Configuration and Package Files

Prepare the Azure Cloud Services (Extended Support) service definition *(.csdef)*, service configuration *(.cscfg)*, and service package *(.cspkg)* files. These can be prepared by:

1. Locating the Azure Cloud Services (Extended Support) resources deployed in the source region, finding the corresponding Azure Storage account, and downloading the files from the storage account container.

2. In Visual Studio, finding the corresponding Azure Cloud Services (Extended Support) project and generating the files using the “Package” feature.

Ensure version control and verify that the current file versions match those deployed in the production environment.

Create Cloud Services in Target Region

Create the Azure Cloud Services (Extended Support) resources in the target Azure region using one of the following methods:

1. Deploy Azure Cloud Services (Extended Support) via Azure portal. See Deploy Cloud Services (Extended Support) using the Azure portal.

2. Deploy Azure Cloud Services (Extended Support) via Azure PowerShell. See Deploy Cloud Services (Extended Support) using Azure PowerShell.

3. Deploy Azure Cloud Services (Extended Support) via ARM template. See Deploy Cloud Services (Extended Support) using ARM Template.

4. Deploy Azure Cloud Services (Extended Support) via SDK. See Deploy Cloud Services (Extended Support) using Azure SDK.

Validation and Testing

1. Start the newly created Azure Cloud Services (Extended Support) and verify its status.

```
az cloud-service show `

--resource-group <ResourceGroupName> `

--cloud-service-name <CloudServiceName>
```

2. Access the Azure Cloud Services (Extended Support) to ensure it is running correctly, for example, by accessing the service URL via a browser.

Clean Up Resources

After confirming that the new Azure Cloud Services (Extended Support) is running correctly, delete the Azure Cloud Services (Extended Support) and related resources from the source region.

```
az cloud-service delete `

--resource-group <ResourceGroupName> `
```

`--cloud-service-name <CloudServiceName>`

Summary

By following the above steps, you have successfully migrated Azure Cloud Services (Extended Support) from one region to another. Ensure to carefully check each step during the migration process to avoid data loss or service disruption. If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

For more information about Azure Cloud Services (Extended Support), please refer to the related documentation:

- Azure Cloud Services (classic) Documentation
- Migrate Azure Cloud Services (classic) to Azure Cloud Services (Extended Support)
- Azure Cloud Services (Extended Support) Documentation

If you have any questions, please contact your Azure support team.

Migrate Azure Application Gateway

This article describes the recommended method for migrating an Application Gateway between Azure regions.

Note: The redeployment steps in this document apply only to the Application Gateway itself and not to the backend services to which the Application Gateway rules route traffic.

Note: If the V1 instance deployed in China North needs to be migrated to another region, upgrade to V2.

Prerequisites

- Confirm that the Azure subscription allows the creation of the Application Gateway SKU in the target region.
- Understand all the services required by the Application Gateway before planning a migration strategy. You must choose an appropriate migration strategy for the services involved in the migration.
  - Ensure that the Application Gateway subnet in the target location has enough address space to accommodate the number of instances required to handle the maximum expected traffic.
- For the deployment of the Application Gateway, you must consider and plan for the setup of the following sub-resources:
  - Frontend configurations (Public/Private IP)
  - Backend pool resources (e.g., VMs, VM Scale Sets, Azure App Service)
  - Private Link
  - Certificates
  - Diagnostic settings
  - Alert notifications
- Ensure that the Application Gateway subnet in the target location has enough address space to accommodate the number of instances required to handle the maximum expected traffic.

Redeployment

To migrate an Application Gateway, you must create a separate Application Gateway deployment in the target location using a new public IP address. Then migrate the workload from the original Application Gateway setup to the new setup. Because you will change the public IP address, DNS configuration, virtual networks, and subnets need to be changed.

If you are migrating solely to obtain availability zone support, refer to Migrate Application Gateway and WAF to Availability Zone Support.

To create a separate Application Gateway, WAF (optional), and IP address:

1. Go to Azure Portal.
2. If you use TLS termination for the key vault, follow the migration procedure for Key Vault. Ensure the Key Vault is in the same subscription as the Application Gateway being migrated. You can either create new certificates or use existing ones for the Application Gateway being migrated.
3. Before the migration, confirm that the virtual network has been migrated. For information on how to migrate a virtual network, refer to Migrate Azure Virtual Network.
4. Before the migration, confirm that the backend pool servers or VMs, VM Scale Sets, PaaS, and other services have been migrated.
5. Create the Application Gateway and configure new frontend public IP addresses for the virtual network:

   - Without WAF: Create Application Gateway
   - With WAF: Create Application Gateway with Web Application Firewall
6. If there are WAF configurations or WAF policies limited to custom rules, Convert configurations or policies to full WAF policies.

7. If your web application using Azure Firewall and Application Gateway employs a Zero Trust Network (source zone), follow the guidance and strategies in Zero Trust Network for Web Applications with Azure Firewall and Application Gateway.

8. Confirm that the Application Gateway and WAF are working correctly.

9. Migrate the configuration to the new public IP address.

    1. Switch public endpoints and private endpoints to point to the new Application Gateway.
    2. Migrate DNS configurations to the new public and/or private IP addresses.
    3. Update endpoints in the consumer applications/services. Typically, this involves changing properties and redeploying the consumer applications/services. However, if you are using new hostnames in the old region deployment, make sure to follow this method.
10. Delete the Application Gateway and WAF resources from the source region.

Migrate Azure ExpressRoute

Overview

Currently, migrating Azure ExpressRoute instances across Azure regions is not supported. For cross-cloud type migrations, we recommend that you create new ExpressRoute circuits and new ExpressRoute gateways in the target Azure region.

Prerequisites

Before proceeding with Azure ExpressRoute migration, please ensure you have completed the following prerequisites:

1. **Network Topology Diagram**:
   - Prepare the current and target network topology diagrams to better plan the migration process.
2. **Confirm Service Provider Support**
   - Confirm whether your service provider supports ExpressRoute services in the target Azure region. Some service providers may offer services only in specific regions or have different coverage areas. > The standard ExpressRoute SKU does not support connections across geopolitical regions. You need to enable the ExpressRoute premium add-on to support global connectivity. Connecting from Azure operated by 21Vianet to other Azure cloud environments is not supported. If needed, please contact the connection service provider.
3. **Notify Stakeholders**:
   - Notify all affected users and teams, and schedule a migration window.
4. **Test Environment**:
   - Simulate the migration process in a test environment to ensure everything goes smoothly.

Related Reference Documents

- Refresh your knowledge by completing the ExpressRoute documentation.
- Learn how to create a new ExpressRoute gateway.
- Read about ExpressRoute connectivity partners and peering locations.
- Read about About ExpressRoute virtual network gateways.

Migrate Azure Load Balancer

Table of Contents

- Introduction
- Prerequisites
- Preparation
- Migration Steps
  - Export Public IP Address Template and Deploy to New Region
  - Export Load Balancer Template and Deploy to New Region
  - Verify and Test
  - Clean Up Resources
- Summary

Introduction

In some situations, you may need to migrate or replicate an Azure load balancer from one region to another. For example, you might want to create another load balancer with the same configuration for testing. You might also want to move the load balancer to another region as part of disaster recovery planning.

Currently, migrating load balancer instances across Azure regions is not supported. However, you can use the Azure Resource Manager template to export the existing configuration of the load balancer and public IP address. Then, deploy to another region in the Azure portal by exporting the load balancer and public IP to a template, modifying parameters for the target region, and then deploying the template to the new region.

Prerequisites

- Ensure the source Azure load balancer is located in an Azure region.
- Load balancers cannot be moved between Azure regions. The new load balancer must be associated with resources in the target region.
- To export Azure load balancer configurations and deploy templates to create the Azure load balancer in another region, you need to have the “Network Contributor” role or a higher role.
- Identify the source network layout and all resources currently in use. This layout includes but is not limited to load balancers, network security groups, public IPs, and virtual networks.
- Verify that the Azure subscription allows the creation of load balancers in the target region. Contact support to enable the required quota if needed.
- Ensure the subscription provides enough resources to support adding a load balancer.
- If you are currently using Azure Load Balancer - Basic, it is recommended to upgrade to Azure Load Balancer - Standard, Please refer to upgrading from basic Load Balancer - Guidance.

Preparation

Before you start, ensure you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Azure CLI installed and configured.
4. Ensure your load balancer has no ongoing tasks and that related resources are backed up.

Migration Steps

Export Public IP Address Template and Deploy to New Region

1. Sign in to the Azure portal, and then select “All resources”.
2. Locate the “Public IP address” used by the load balancer and open it.
3. Select “Settings” > “Export template”.
4. Select “Deploy” at the top of the “Export template” page.
5. Choose “Edit parameters” to open the *parameters.json* file in the online editor.
6. To edit the “Public IP address” name, change the **value** property under **parameters** to the name of the target public IP address, and enclose the name in quotes. After modifying, select “Save” in the editor.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#"
,
  "contentVersion": "1.0.0.0",
  "parameters": {
    "publicIPAddresses_myVM1pubIP_name": {
    "value": "<target-publicip-name>"
     }
    }
    }
```

7. Choose “Edit template” to open the *template.json* file in the online editor.
8. To edit the target region where the public IP will be moved, change the **location** property value under **resources**.

```
"resources": [
    {
    "type": "Microsoft.Network/publicIPAddresses",
    "apiVersion": "2019-06-01",
    "name": "[parameters('publicIPAddresses_myPubIP_name')]",
    "location": "<target-region>",
    "sku": {
        "name": "Standard",
        "tier": "Regional"
    },
    "properties": {
        "provisioningState": "Succeeded",
        "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
        "ipAddress": "52.177.6.204",
```

```
        "publicIPAddressVersion": "IPv4",
        "publicIPAllocationMethod": "Static",
        "idleTimeoutInMinutes": 4,
        "ipTags": []
      }
      }
    ]
```

To get regional location codes, refer to Azure Locations. Region codes are region names without spaces.

9. You can also change other template parameters as needed:

   – **SKU**: The SKU of the “Public IP address” can be changed from standard to basic or from basic to standard by changing the **name** property under **sku** in the *template.json* file:

```
"resources": [
{
  "type": "Microsoft.Network/publicIPAddresses",
  "apiVersion": "2019-06-01",
  "name": "[parameters('publicIPAddresses_myPubIP_name')]",
  "location": "<target-region>",
  "sku": {
    "name": "<Standard/Basic>",
    "tier": "Regional"
  },
```

   – **Availability zone**: The zone attribute can be changed to alter the zone of the “Public IP address”. If the zone attribute is not specified, the public IP will be created without a zone. A single zone can be specified to create a zonal public IP or all three zones to create a zone-redundant public IP.

```
"resources": [
{
  "type": "Microsoft.Network/publicIPAddresses",
  "apiVersion": "2019-06-01",
  "name": "[parameters('publicIPAddresses_myPubIP_name')]",
  "location": "<target-region>",
  "sku": {
    "name": "Standard",
    "tier": "Regional"
  },
  "zones": [
    "1",
    "2",
    "3"
  ],
```

   – **Public IP allocation method** and **idle timeout**: The IP allocation method can be changed by changing the **publicIPAllocationMethod** property value from “Static” to “Dynamic,” or from “Dynamic” to “Static”. The idle timeout value can be changed by modifying the **idleTimeoutInMinutes** property value. The default value is 4. Select “Save” in the online editor.

```
"resources": [
{
```

```
    "type": "Microsoft.Network/publicIPAddresses",
    "apiVersion": "2019-06-01",
    "name": "[parameters('publicIPAddresses_myPubIP_name')]",
    "location": "<target-region>",
    "sku": {
        "name": "Standard",
        "tier": "Regional"
    },
    "zones": [
        "1",
        "2",
        "3"
    ],
    "properties": {
        "provisioningState": "Succeeded",
        "resourceGuid": "7549a8f1-80c2-481a-a073-018f5b0b69be",
        "ipAddress": "52.177.6.204",
        "publicIPAddressVersion": "IPv4",
        "publicIPAllocationMethod": "Static",
        "idleTimeoutInMinutes": 4,
        "ipTags": []
```


10. Select “Save” in the online editor.
11. Choose “Basics” > “Subscription” to select the subscription where the “Public IP address” will be deployed.
12. Choose “Basics” > “Resource group” to select the resource group where the “Public IP address” will be deployed. Select “Create new” to create a new resource group for the target public IP. Make sure the selected name is different from the source resource group of the existing source public IP.
13. Confirm that “Basics” > “Region” is set to the correct target region.
14. Under “Settings,” confirm that the “Public IP address” name matches the name entered earlier in the *parameters* editor.
15. Select “Review and create”.
16. Choose “Create” to deploy the target “Public IP address”.
17. If another public IP is used for the outbound NAT of the load balancer to be moved, repeat the above steps to export the second outbound public IP and deploy it to the target region.

Export Load Balancer Template and Deploy to New Region

1. Sign in to the Azure portal, and then select “All resources”.
2. Locate the “Load balancer” you want to migrate and select it.
3. Select “Settings” > “Export template”.
4. Select “Deploy” under “Export template”.
5. Choose “Edit parameters” to open the *parameters.json* file in the online editor.

6. To edit the load balancer name parameter, change the **value** property of the load balancer name to the desired name. Enclose the name in quotes.

```
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "loadBalancers_myLoadbalancer_ext_name": {
    "value": "<target-external-lb-name>"
},
    "publicIPAddresses_myPubIP_in_externalid": {
    "value": "<target-publicIP-resource-ID>"
},
```

7. To edit the load balancer's public IP address to the value of the public IP address created in the previous steps, you need to first obtain the resource ID of that public IP address and then paste it into the *parameters.json* file. Steps to get the resource ID:

    1. In another browser tab or window, sign in to the Azure portal and select "Resource groups".
    2. Locate the public IP address created in the previous steps, and open that resource.
    3. Select "Settings" > "Properties".
    4. On the right, highlight "Resource ID" and copy it to the clipboard. Alternatively, you can select "Copy to clipboard" to the right of the resource ID path.
    5. Paste the resource ID into the **value** property value in the "Edit parameters" editor opened in another browser window or tab:

```
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "loadBalancers_myLoadbalancer_ext_name": {
    "value": "<target-external-lb-name>"
},
    "publicIPAddresses_myPubIP_in_externalid": {
    "value": "<target-publicIP-resource-ID>"
},
```

    6. Select "Save".

8. If the load balancer is configured with outbound NAT and outbound rules, a third entry will appear in this file for the external ID of the outbound public IP. Repeat the above steps in the **target region** to get the resource ID of the outbound public IP. Paste the resource ID into the *parameters.json* file:

```
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "loadBalancers_myLoadbalancer_ext_name": {
        "value": "<target-external-lb-name>",

    },
```

```
        "publicIPAddresses_myPubIP_in_externalid": {
        "value": "<target-publicIP-resource-ID>",

    },
        "publicIPAddresses_myPubIP_out_externalid": {
        "defaultValue": "<target-publicIP-outbound-resource-ID>",

    }
  },
```

9. Choose “Template” > “Edit template” to open the *template.json* file in the online editor.

10. To edit the target region where the external load balancer configuration will be moved, change the **location** property under **resources** in *template.json* file:

    To get regional location codes, refer to Azure Locations. Region codes are region names without spaces.

```
"resources": [
    {
        "type": "Microsoft.Network/loadBalancers",
        "apiVersion": "2019-06-01",
        "name": "[parameters('loadBalancers_myLoadBalancer_name')]",
        "location": "<target-external-lb-region>",
        "sku": {
            "name": "Standard",
            "tier": "Regional"
        },
```

11. You can also change other template parameters according to your needs and requirements:

    – **SKU**: The SKU of the “Load balancer” can be changed from “Standard” to “Basic,” or from “Basic” to “Standard,” by changing the name attribute under **sku** in the *template.json* file:

```
"resources": [
{
  "type": "Microsoft.Network/loadBalancers",
  "apiVersion": "2019-06-01",
  "name": "[parameters('loadBalancers_myLoadBalancer_name')]",
  "location": "<target-external-lb-region>",
  "sku": {
      "name": "Standard",
      "tier": "Regional"
  },
```

    – **Load balancing rules**: Load balancing rules can be added or removed from the configuration by adding or removing entries in the **loadBalancingRules** section of the *template.json* file:

```
"loadBalancingRules": [
        {
            "name": "myInboundRule",
            "etag": "W/\"39e5e9cd-2d6d-491f-83cf-b37a259d86b6\"",
```

```
            "properties": {
              "provisioningState": "Succeeded",
              "frontendIPConfiguration": {
                "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_name')), '/frontendIPConfigurations/myfrontendIPinbound')]"
              },
              "frontendPort": 80,
              "backendPort": 80,
              "enableFloatingIP": false,
              "idleTimeoutInMinutes": 4,
              "protocol": "Tcp",
              "enableTcpReset": false,
              "loadDistribution": "Default",
              "disableOutboundSnat": true,
              "backendAddressPool": {
                "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_name')), '/backendAddressPools/myBEPoolInbound')]"
              },
              "probe": {
                "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_name')), '/probes/myHTTPProbe')]"
              }
            }
          }
        ]
```

– **Probes**: Probes can be added or removed from the load balancer configuration by adding or removing entries in the **probes** section of the *template.json* file:

```
"probes": [
        {
          "name": "myHTTPProbe",
          "etag": "W/\"39e5e9cd-2d6d-491f-83cf-b37a259d86b6\"",
          "properties": {
            "provisioningState": "Succeeded",
            "protocol": "Http",
            "port": 80,
            "requestPath": "/",
            "intervalInSeconds": 15,
            "numberOfProbes": 2
          }
        }
      ],
```

– **Inbound NAT rules**: Inbound NAT rules can be added or removed for the load balancer by adding or removing entries in the **inboundNatRules** section of the *template.json* file:

```
"inboundNatRules": [
        {
```

```
            "name": "myInboundNATRule",
            "etag": "W/\"39e5e9cd-2d6d-491f-83cf-b37a259d86b6\"",
            "properties": {
              "provisioningState": "Succeeded",
              "frontendIPConfiguration": {
                "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalanc
ers_myLoadBalancer_name')), '/frontendIPConfigurations/myfrontendIPinbound')]"
              },
              "frontendPort": 4422,
              "backendPort": 3389,
              "enableFloatingIP": false,
              "idleTimeoutInMinutes": 4,
              "protocol": "Tcp",
              "enableTcpReset": false
            }
          }
        ]
```

To complete the addition or removal of an inbound NAT rule, it must appear as a **type** property at the bottom of the *template.json* file, or confirm it was removed:

```
{
  "type": "Microsoft.Network/loadBalancers/inboundNatRules",
  "apiVersion": "2019-06-01",
  "name": "[concat(parameters('loadBalancers_myLoadBalancer_name'), '/myInboundNATRule')]",
  "dependsOn": [
    "[resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_
name'))]"
  ],
  "properties": {
    "provisioningState": "Succeeded",
    "frontendIPConfiguration": {
      "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_my
LoadBalancer_name')), '/frontendIPConfigurations/myfrontendIPinbound')]"
    },
    "frontendPort": 4422,
    "backendPort": 3389,
    "enableFloatingIP": false,
    "idleTimeoutInMinutes": 4,
    "protocol": "Tcp",
    "enableTcpReset": false
  }
}
```

– **Outbound rules**: Outbound rules can be added or removed from the configuration by editing the **outboundRules** property in *template.json* file:

```
"outboundRules": [
          {
```

```
                "name": "myOutboundRule",
                "etag": "W/\"39e5e9cd-2d6d-491f-83cf-b37a259d86b6\"",
                "properties": {
                  "provisioningState": "Succeeded",
                  "allocatedOutboundPorts": 10000,
                  "protocol": "All",
                  "enableTcpReset": false,
                  "idleTimeoutInMinutes": 15,
                  "backendAddressPool": {
                    "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_name')), '/backendAddressPools/myBEPoolOutbound')]"
                  },
                  "frontendIPConfigurations": [
                    {
                      "id": "[concat(resourceId('Microsoft.Network/loadBalancers', parameters('loadBalancers_myLoadBalancer_name')), '/frontendIPConfigurations/myfrontendIPoutbound')]"
                    }
                  ]
                }
              }
            ]
```

12. In the editor, select “Save.”

13. Select “Basic Information” > “Subscription” to choose the subscription to which you want to deploy the load balancer.

14. Select “Basic Information” > “Resource Group” to choose the resource group to which you want to deploy the load balancer. You can select “Create New” to create a new resource group for the target external load balancer. Ensure that the resource group name is different from the source resource group name of the source load balancer.

15. Confirm that “Basic Information” > “Region” is set to the target region for the load balancer deployment.

16. Under “Settings,” confirm that the name matches the name previously entered in the parameters editor. Confirm that the resource IDs for all public IPs in the configuration are filled in.

17. Select “Review and Create.”

18. Select “Create” to complete the resource deployment of the load balancer.

Validation and Testing

1. Start and verify the status of the newly created load balancer.

```
az network lb show `
--resource-group <TargetResourceGroupName> `
--name <TargetLoadBalancerName>
```

2. Test if the new load balancer configuration is working properly.

Clean Up Resources

If the new load balancer is confirmed to be working properly, you can delete the old load balancer and related res ources in the source region.

```
az network lb delete `
--resource-group <ResourceGroupName> `
--name <LoadBalancerName>
```

Summary

By following the above steps, you have successfully migrated a load balancer from one region to another. Ensure to carefully check each step during the migration to avoid data loss or service interruption. If you encounter any i ssues, you can refer to the official Azure documentation or contact Azure technical support.

Learn how to migrate an Azure Load Balancer. Relevant documentation:

- Azure Load Balancer Overview
- Azure Load Balancer Tutorial
- Create a New Azure Load Balancer
- Upgrade Azure Load Balancer from “Basic” to “Standard”

If you have any questions, please contact your Azure support team.

Migrate Azure Network Watcher

Introduction

Migrating a Network Watcher instance across Azure regions isn’t supported at this time.
Network Watcher is a service enabled for a specific Azure region to monitor and diagnose network traffic.
We recommend that you create and configure a new Network Watcher instance in the target region. Afterward, compare results between the old and new environments.

Summary

Before migrating in a production environment, test and validate in a test environment.
If you encounter any issues, refer to Azure official documentation or contact Azure technical support.

For more information: * Review the Network Watcher Overview. * Learn more about enabling or disabling Azure Network Watcher. * Learn more about flow logging for network security groups. * Read about the Connection Monitor. * Check out the Network Watcher FAQ. —

Migrate Azure Virtual Network

Table of Contents

- Introduction
- Prerequisites
- Preparations
- Migration Steps
  - Export Template
  - Modify Template
  - Create Virtual Network
  - Configure Virtual Network
  - Migrate Associated Resources
  - Validation and Testing
  - Cleanup Resources
- Summary

Introduction

Currently, cross-region migration of virtual networks is not supported in Azure. We recommend creating new virtual networks in the target region and migrating resources to these virtual networks.

This guide aims to provide instructions on how to migrate an Azure Virtual Network from one region to another. It describes how to achieve cross-region migration of a virtual network by redeploying it in the new region. Redeployment supports both independent migrations of multiple workloads and changes to the private IP address range in the target region. It is recommended to use ARM Resource Manager templates for the migration.

Prerequisites

- Identify all resources associated with the virtual network to be migrated.
- Network peering
  - Load balancer
  - User-Defined Routes (UDR)
  - NAT Gateway
  - DDOS Protection Plan
  - Network Security Groups (NSG)
  - Reserved Private IP Address (Public Static IP Address)
  - Application Security Groups (ASG)
- Confirm that the virtual network is located in the source Azure region.
- To export and deploy a virtual network template to create a virtual network in another region, the “Network Contributor” role or a higher-level role is required.
- Identify the source network layout and all currently used resources. This layout includes, but is not limited to, load balancers, Network Security Groups (NSG), and public IP addresses.
- Verify if the Azure subscription allows creating virtual networks in the target region. Contact support to enable the necessary quotas.
- Understand the following considerations:

- Enabling private IP address range changes allows the separate redeployment of multiple workloads within the virtual network.
- The redeployment method supports options to enable and disable private IP address range changes in the target region.
- If private IP changes are not enabled in the target region, data migration scenarios requiring communication between the source and target regions can only be established using public endpoints (public IP addresses).

Preparations

Before you start, ensure you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Azure CLI installed and configured.
4. The diagnostic storage account containing Network Watcher NSG logs moved.
5. The Network Security Group (NSG) relocated.
6. DDoS Protection Plan disabled.

Migration Steps

Export Template

Export the ARM template that contains your Virtual Network settings and information.

1. Sign in to the Azure Portal.
2. Select “All resources”, then select your Virtual Network.
3. Select “Settings” > “Export template”.
4. Select the “Include parameters” checkbox.
5. Select “Download” on the “Export template” page.
6. Locate the .zip file downloaded from the Azure portal, and extract it locally.

   This zip file contains two files: *template.json* and *parameters.json*, which form the template.

Modify Template

Load and modify the template to create a new Virtual Network in the target region.

1. In the Azure portal, select “Create a resource”.
2. In “Search services and marketplace”, type “Template deployment” and press **ENTER**.
3. Select “Template deployment (deploy using custom templates)”.
4. Select “Create”.
5. Select “Build your own template in the editor”.

6. Select “Load file”, then select the “*template.json*” file downloaded in the previous section.

7. Replace “<*target-virtual-network-name*>” in the following code with the target Virtual Network name.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#"
,
  "contentVersion": "1.0.0.0",
  "parameters": {
    "virtualNetworks_myVNET1_name": {
      "value": "<target-virtual-network-name>"
    }
  }
}
```

8. Replace “<*target-region*>” in the following code with the target region.

```
"resources": [
        {
            "type": "Microsoft.Network/virtualNetworks",
            "apiVersion": "2019-06-01",
            "name": "[parameters('virtualNetworks_myVNET1_name')]",
            "location": "<target-region>",
            "properties": {
                "provisioningState": "Succeeded",
                "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
                "addressSpace": {
                    "addressPrefixes": [
                        "10.0.0.0/16"
                    ]
                },
```

9. You can also change other parameters in the template as needed:

- **Address Space**: Modify the resources > addressSpace section and change the addressPrefixes property to change the address space of the Virtual Network.

```
"resources": [
        {
        "type": "Microsoft.Network/virtualNetworks",
        "apiVersion": "2019-06-01",
        "name": "[parameters('virtualNetworks_myVNET1_name')]",
        "location": "<target-region>",
        "properties": {
        "provisioningState": "Succeeded",
        "resourceGuid": "6e2652be-35ac-4e68-8c70-621b9ec87dcb",
        "addressSpace": {
            "addressPrefixes": [
            "10.0.0.0/16"
            ]
```

- **Subnet**: You can change the subnet name and subnet address space or add contents by modifying the subnets section of the template. You can change the subnet name by modifying the **name** attribute. You can change the subnet address space by changing the **addressPrefix** attribute.


```
"subnets": [
        {
        "name": "subnet-1",
        "etag": "W/\"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea\"",
        "properties": {
        "provisioningState": "Succeeded",
        "addressPrefix": "10.0.0.0/24",
        "delegations": [],
        "privateEndpointNetworkPolicies": "Enabled",
        "privateLinkServiceNetworkPolicies": "Enabled"
        }
        },
        {
        "name": "GatewaySubnet",
        "etag": "W/\"d9f6e6d6-2c15-4f7c-b01f-bed40f748dea\"",
        "properties": {
        "provisioningState": "Succeeded",
        "addressPrefix": "10.0.1.0/29",
        "serviceEndpoints": [],
        "delegations": [],
        "privateEndpointNetworkPolicies": "Enabled",
        "privateLinkServiceNetworkPolicies": "Enabled"
        }
        }
    ]
```


10. After making the changes, select “Save” below the “*template.json*” file.

Create Virtual Network

1. Enter or select property values:
    - **Subscription**: Select the Azure subscription.
    - **Resource group**: Select the resource group for the target region.
    - **Region**: Select the Azure region where you want to move the virtual network.
2. Select “Review + create”, then “Create”.

Configure Virtual Network

Certain configurations within the virtual network are not exported into the template and must be reconfigured in the new virtual network, including but not limited to the following items:

- Azure Firewall
- Peering
- Service Endpoints

- Private Endpoints

Make sure to configure the settings in the new virtual network as required. Refer to the source virtual network to understand how to configure these features.

Migrate Associated Resources

Change the configurations of the virtual machines and other network-related resources to associate them with the new virtual network, including but not limited to:

- Virtual machines
- DNS Servers
- Network Security Groups

Refer to the relevant resource migration manual for the migration method.

Validation and Testing

1. Confirm that all resources are deployed in the new virtual network and that all configurations are correct.
2. Verify that the virtual machines can start correctly and that the configurations are correct.
3. Validate the performance of the applications in the new network to ensure they are running correctly.

Cleanup Resources

Once all resources are confirmed to be running correctly in the new region, you can delete the old resources:

```powershell
# Delete the virtual network in the source region
az network vnet delete `
--resource-group <ResourceGroupName> `
--name <VNetName>
```

Summary

By following the above steps, you have successfully migrated an Azure Virtual Network from one region to another. Ensure you double-check each step during the migration process to avoid data loss or service interruptions. If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

To learn more about Virtual Network migration, refer to the following documents:

- Overview of Azure Virtual Network
- Azure Virtual Network Tutorials
- Planning for Azure Virtual Network

If you have any questions, please contact your Azure support team.

Migrate Virtual WAN

Overview

This guide will help you migrate Azure Virtual WAN from one region to another. Please note that Azure Virtual WAN itself does not support direct regional migration, so resources need to be manually recreated and configured.

In most cases, since resources are closely related to the network, Azure Virtual WAN will impact the usage of these resources during the migration process. A relatively smooth migration scheme is to create a new Azure Virtual WAN in the new region and interconnect it with the current virtual WAN. If the virtual networks (VNet) under the current virtual WAN are located in the region to be migrated, you can create new virtual networks in other regions. Detailed migration plans can be found in the Azure Virtual Network Region Migration Guide. Post migration, you can connect them to the new Azure Virtual WAN center. Detailed steps can be found in Connect Virtual Network to Virtual WAN Hub. If the original virtual networks are not in the region to be migrated, you need to disconnect from the current virtual WAN hub and reconnect to the new Azure Virtual WAN.

> The migration of Azure Virtual WAN will cause temporary network interruptions, so planning and preparation are necessary in advance.

Migration Preparation

- **Assess and Backup Existing Configuration**
  - **Assess Existing Virtual WAN Configuration**: Perform a comprehensive check of the current virtual WAN settings, including virtual networks (VNet), connections, route tables, VPN gateways, and ExpressRoute gateways.
  - **Backup Current Configuration**: Ensure all configuration items are backed up so that they can be quickly restored in case of issues during migration. Pay special attention to backing up detailed information of virtual networks, connections, route tables, VPN gateways, and ExpressRoute gateways.
- **Check Target Region Support**
  - Confirm that the target region supports all required virtual WAN features and resource types. For more details, please refer to the article Virtual WAN Partners and Locations.
- **Assess the Impact of Network Interruptions on Business**
  - **Business Impact Analysis**: Identify which business applications and services rely on the current virtual WAN and assess their potential impact during the migration. Determine the potential downtime and consequences for critical business applications.
  - **Stakeholder Communication**: Communicate with all relevant teams and stakeholders (such as business departments, IT operations teams, security teams, etc.) to ensure they understand the migration plan and possible network downtime.
  - **User Notification**: Notify all potentially affected users and customers in advance, informing them of the migration schedule and possible service interruptions.
- **Formulate Contingency Measures**
  - **Migration Window Selection**: Choose a low business peak period for migration to minimize the impact on business operations.
  - **Temporary Solutions**: Prepare temporary solutions or backup paths for critical business applications and services. For example, setting up temporary VPN connections or using other network services to ensure continuous operation of critical businesses.
  - **Rollback Plan**: Develop a detailed rollback plan to address any unforeseen issues that might arise during the migration, ensuring a quick return to the original state if needed.
- **Technical Preparation**

- **Test Migration Steps**: Simulate the migration process in a test environment to verify the accuracy of all steps and configurations. Ensure the new region's virtual WAN and related resources work correctly.
- **Monitoring and Logging**: Set up detailed monitoring and logging to monitor network status and performance in real-time during the migration. Promptly identify and resolve potential issues.

Create a New Virtual WAN and VPN Connection

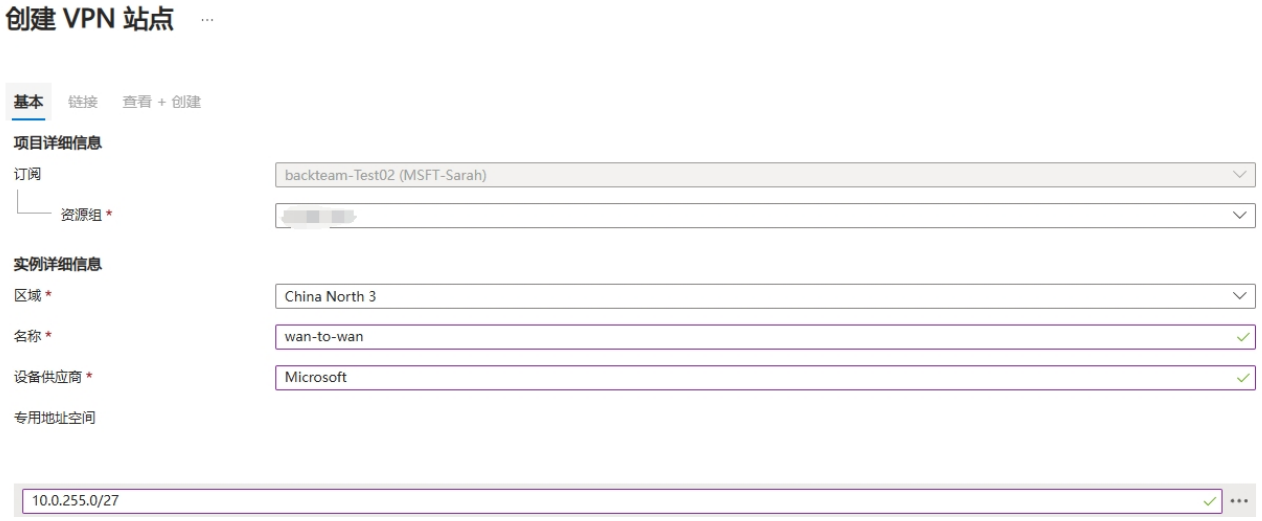
The VPN connection between two virtual WANs is an optional method rather than a mandatory solution.

1. Create a new Azure Virtual WAN in the new region

2. Create a new Hub

3. Configure Site-to-Site Gateway

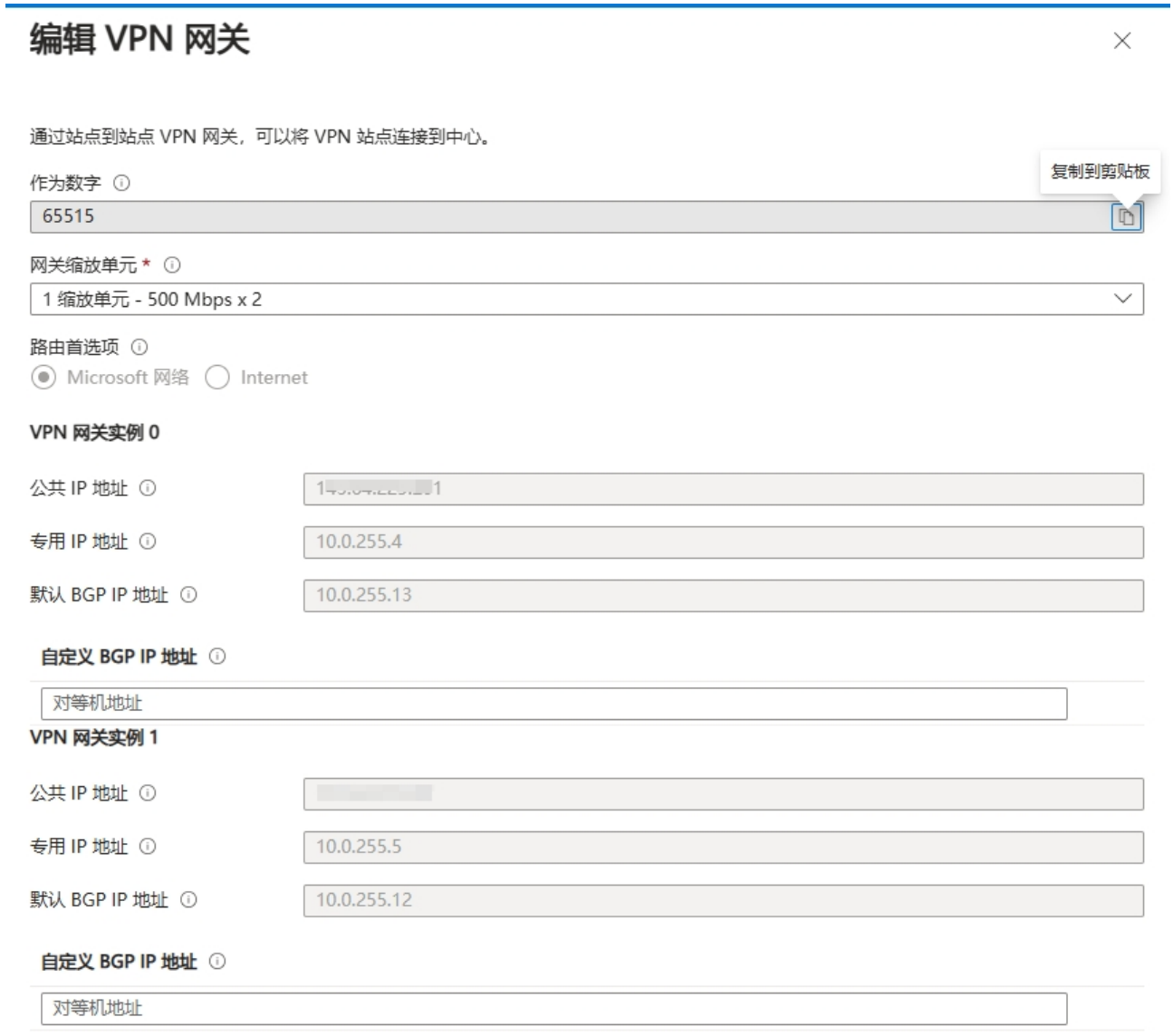
4. Create VPN Site



5. Add Link. If the original virtual WAN did not create a VPN Gateway, you can write an unassigned IP address here, we will modify this configuration later.

6. Connect VPN Site in the Hub

7. After creation, obtain the VPN Gateway Public IP address



8. Repeat the above steps to create VPN Sites and connections in the existing Azure Virtual WAN.
9. [Optional] If the correct “Link IP address” was not used when creating the VPN link earlier, you can correct this address after adding it to the original WAN. Check the connection status after completion.

10. Test the network connectivity between the two WANs

Migrate Original Region VNET

- Azure Virtual Network Region Migration Guide
- Connect Virtual Network to Virtual WAN Hub

Migrate Non-original Region VNET to the New Virtual WAN Hub

This operation will cause network interruptions, please prepare before migration

1. Delete from the existing Hub

2. Add to the new Hub

Relevant Reference Documents

- Virtual WAN Documentation
- Tutorial: Create Site-to-Site Connection using Azure Virtual WAN
- Tutorial: Create P2S User VPN Connection using Azure Virtual WAN
- Tutorial: Create ExpressRoute Association with Virtual WAN

Migrate Azure NAT Gateway

Overview

Azure NAT Gateway is a fully managed and highly resilient Network Address Translation (NAT) service. You can use Azure NAT Gateway to provide outbound connectivity to the Internet for all instances in a private subnet while keeping it completely private.

NAT Gateway instances cannot be directly moved from one region to another. A workaround is to use the Azure Resource Mover to migrate all resources associated with the existing NAT Gateway to the new region. Then, create a new instance of the NAT Gateway in the new region and associate the moved resources with the new instance. Once the new NAT Gateway is operational in the new region, you can delete the old instance in the previous region.

Downtime

To understand potential downtime involved, refer to Cloud Adoption Framework for Azure: Select a relocation method.

Related References

- Create and configure a NAT Gateway after moving resources to another region
- Quickstart: Create a NAT Gateway using the Azure portal
- NAT Gateway and Availability Zones
- Designing Virtual Networks with Azure NAT Gateway

Migrate Azure VPN Gateway

Overview

Migrating Azure VPN Gateway instances across Azure regions is currently not supported. We recommend creating and configuring new instances of the VPN Gateway in the new region.
You can use the portal or PowerShell to collect information about the current VPN Gateway configuration. In PowerShell, use the set of cmdlets starting with `Get-AbureRmVirtualNetworkGateway`.
Be sure to update your local configuration. Additionally, after updating the Azure network environment, delete any existing rules in the old IP address range.

Prerequisites

Before migrating a VPN Gateway, ensure that the following prerequisites have been completed:

1. **Backup Configuration**:
   - Ensure to backup all configurations of the current VPN Gateway, including IP addresses, subnets, route tables, and security groups.
2. **Access Permissions**:
   - Ensure you have sufficient permissions to create, modify, and delete VPN Gateways and related resources.
3. **Network Topology Diagram**:
   - Prepare both the current and target network topology diagrams to better plan the migration process.
4. **Notify Stakeholders**:
   - Inform all affected users and teams and schedule a migration window.
5. **Test Environment**:
   - Simulate the migration process in a test environment to ensure everything goes smoothly.

Downtime Impact

During the migration process, the following downtime impacts may occur:

1. **Network Interruption**:
   - There may be a brief network interruption when switching to the new VPN Gateway.
2. **Service Unavailability**:
   - Applications and services relying on the VPN connection may be unavailable during the migration process.
3. **User Connection Interruption**:
   - Remote users' VPN connections may be interrupted during the migration and require reconnection.

Related Reference Documents

- Refresh your knowledge by completing the VPN Gateway tutorial.
- Learn how to Tutorial: Create a site-to-site VPN connection in the Azure portal.
- Check out the Get-AzureRmVirtualNetworkGateway PowerShell cmdlet.
- Review the az network vnet commands.
- Read the blog post: Creating a site-to-site connection.
- Check the Azure VPN Gateway documentation.

Migrate Web Application Firewall (WAF)

Overview

To migrate the Application Gateway and WAF (optional), you must create a separate Application Gateway deployment in the target location **using a new public IP address**. Then, migrate the workload from the original Application Gateway setup to the new setup.

Prerequisites

- Verify that your Azure subscription allows the creation of Application Gateway SKU in the target region.
- Understand all services required by the Application Gateway before planning the migration strategy. You must choose an appropriate migration strategy for the services involved in the migration.
  - Ensure that the Application Gateway subnet at the target location has enough address space to accommodate the number of instances required to handle the maximum expected traffic.
- For the Application Gateway deployment, you must consider and plan the setup of the following sub-resources:
  - Frontend configuration (Public/Private IP)
  - Backend pool resources (e.g., VMs, Virtual Machine Scale Sets, Azure App Services)
  - Private links
  - Certificates
  - Diagnostic settings
  - Alert notifications
- Ensure that the Application Gateway subnet at the target location has enough address space to accommodate the number of instances required to handle the maximum expected traffic.

Migration Steps

Refer to Relocate Azure Application Gateway and Web Application Firewall (WAF) to another region for detailed migration steps.

Related Reference Documents

- Azure Application Gateway Documentation

Migrate Azure Bastion Resource

This guide will help you migrate Azure Bastion resources from one region to another. Since Azure Bastion resources do not support direct cross-region migration, you need to manually recreate the resources and update the relevant configurations.

Preparation before Migration

1. **Plan the new network design**: Assess the network architecture of the target region to ensure it suits the new Azure Bastion resources, and determine the virtual network (VNet) and subnet configurations for the target region.
2. **Backup configurations**: Document the current Azure Bastion configurations, including virtual networks (VNet), subnets, network security groups (NSG), etc.
3. **Permissions check**: Ensure you have sufficient permissions to create and manage Azure resources in both the source and target regions.

Migration Steps

1. Create a new Azure Bastion in the target region

Please read Tutorial: Deploy Azure Bastion using specified settings

2. Update the virtual network and subnet configurations

1. Update the virtual network configuration in the target region to accommodate the new Azure Bastion.
2. If needed, update the network security group (NSG) rules to allow necessary traffic.

3. [Optional] Use virtual network peering

If you need to connect original resources using the new Azure Bastion, you can learn about Virtual network peering and Azure Bastion.

4. Test the connection

1. Use the new Azure Bastion resources to connect to virtual machines in the virtual network and ensure the connection is operational.

5. Delete the Azure Bastion resources in the source region

1. After confirming that the Azure Bastion resources in the target region are functioning correctly, you can delete the Bastion resources in the source region to avoid extra charges.
2. Navigate to the Bastion resources in the source region and click **Delete**.

Frequently Asked Questions

1. **Can Azure Bastion resources be migrated directly?**

   No, Azure Bastion resources do not support direct cross-region migration. You must manually recreate and configure in the target region.

2. **How to ensure no data is lost during the migration?**

   Azure Bastion itself does not store data, but ensure that the virtual network and subnet configurations are correct to prevent connection issues.

3. **Can the Public IP be retained after migration?**

   No, Public IP addresses are regional resources and cannot be migrated across regions. Therefore, when creating new Azure Bastion resources in the target region, you need to assign a new Public IP address. Ensure to notify relevant parties about the IP address change before migration to avoid connection interruptions.

Related Reference Documents

Azure Bastion Documentation

Migrate Azure Container Registry

Table of Contents

- Azure Container Registry Regional Migration Guide
  - Table of Contents
  - Introduction
  - Geo-Replication
  - Recreate and Import
    - Prerequisites
    - Service Endpoint Considerations
    - Private Endpoint Considerations
      - Azure Private Endpoint DNS Integration Considerations
    - Relocate to Another Region

Introduction

This guide is intended to help you migrate an Azure Container Registry (ACR) from one region to another. This guide is based on operations using Azure CLI and is specifically for the Azure in China region.

There are two main options for migrating ACR:

1. Geo-Replication
2. Recreate and Import

Geo-Replication

Companies needing local presence or hot backup can choose to run services from multiple Azure regions. Best practices suggest placing a container registry in each region where images run, allowing for near-network operations to achieve fast and reliable image layer transfers. Geo-replication allows an Azure container registry to act as a single registry, providing multi-master regional registries to multiple regions.

Benefits of geo-replication registries:

- A single registry with image and tag names usable across multiple regions
- Improved performance and reliability of regional deployments via near-network registry access
- Reduced data transfer costs by pulling image layers from a locally replicated registry in the container host's region or neighboring region
- Single management of the registry across multiple regions
- Registry resilience during regional outages

For detailed operations, please refer to the official documentation: Geo-replication in Azure Container Registry

Recreate and Import

You might need to move an Azure container registry from one Azure region to another. For example, you run a development pipeline or host a new deployment target in a different region and wish to provide a registry in the nearby region.

Although the Azure Resource Mover cannot automatically move an Azure container registry, you can manually move the container registry to another region:

- Export registry settings to a resource manager template
- Deploy the registry in a different Azure region using the template
- Import the registry contents from the source registry to the target registry

For detailed operations, please refer to the official documentation: Manually Move a Container Registry to Another Region

Prerequisites

Using a manual method to transfer regions has the following prerequisites:

- Registry relocation is only available within the same Active Directory tenant. This restriction applies to registries encrypted and decrypted with customer-managed keys.
- If the source registry has availability zones enabled, the target region must also support availability zones. For more information on availability zone support in Azure Container Registry, see Enable Zone Redundancy in Azure Container Registry for Resiliency and High Availability.

Service Endpoint Considerations

The virtual network service endpoint for Azure container registries restricts access to specified virtual networks. Additionally, these endpoints can restrict access to a range of IPv4 (Internet Protocol version 4) address ranges. Any users connecting to the registry from outside will not have access to these resources. If service endpoints are configured in the source region for the registry resource, the same must be done in the target region. Here are the steps for this scenario:

- To successfully recreate the registry in the target region, VNet and subnets must be created beforehand. If moving all these resources using the Azure Resource Mover tool, service endpoints are not automatically configured, and you must provide manual configuration.
- Secondly, modifications are required in the IaC for Azure Container Registry. In the networkAcl section, under virtualNetworkRules, add rules for the target subnet. Ensure that the ignoreMissingVnetServiceEndpoint flag is set to False; this ensures that without service endpoints configured in the target region, IaC cannot deploy the Azure Container Registry. This ensures that prerequisites are met in the target region.

Private Endpoint Considerations

Azure Private Link provides a private connection from a virtual network to Azure platform as a service (PaaS), customer-owned services, or Microsoft partner services. Private Link simplifies the network architecture and securely connects endpoints in Azure, keeping the data off the public internet.

Azure Private Endpoint DNS Integration Considerations

You must correctly configure DNS settings to resolve the private endpoint's IP address to the fully qualified domain name (FQDN) of the connection string.

Existing Microsoft Azure services may already have DNS configurations for public endpoints. This configuration must be overridden to use the private endpoint for connectivity.

A network interface associated with a private endpoint includes the necessary information for DNS configuration. This information includes FQDN and private IP addresses of the private link resource.

Options for configuring DNS settings for private endpoints:

Using a hosts file (recommended only for testing). You can use hosts files on virtual machines to override DNS. Using Private DNS Zones. You can use private DNS zones to override DNS resolution for the private endpoint.

Private DNS zones can be linked to your virtual network to resolve specific domains. Using a DNS forwarder (optional). You can use a DNS forwarder to override DNS resolution for private link resources. Create DNS forwarding rules to use private DNS zones on DNS servers hosted within the virtual network. The Azure Container Registry must be configured in the target region with the Premium tier.

When public network access to the registry is disabled, some trusted services, including Azure Security Center, require a network setting bypass to access the registry.

If the registry has an approved private endpoint and public network access is disabled, listing repositories and tags from outside the virtual network using the Azure portal, Azure CLI, or other tools is not possible.

For new replicas, new DNS records must be manually added for data endpoints in the target region.

Relocate to Another Region

For a more detailed explanation of considerations, you can refer to the Global Azure documentation on Relocate an Azure Container Registry to Another Region, but note that there are differences between Global Azure and Azure in China; this document is for reference only.

Migrate Azure Service Fabric

Table of Contents

- Introduction
- Preparations
- Migration Steps
  - Create a New Service Fabric Cluster
  - Migrate Applications
  - Redirect Traffic
  - Validation and Testing
  - Clean Up Resources
- Summary

Introduction

Azure Service Fabric cluster resources are intrinsically confined to a single region. Therefore, cluster resources cannot be moved across regions. The current method to achieve “cross-region movement” is as follows: first, create a new cluster in the target region, then migrate the existing applications, and finally redirect the traffic to the new target region. This document outlines the steps required to complete this migration. There are several decision points that can determine the complexity of the migration. These inputs include how the clusters and applications are set up and configured, how communication within the cluster works, and whether the workloads are stateless, stateful, or both.

This manual aims to guide you on how to migrate an Azure Service Fabric cluster from one region to another. We will describe the main migration steps to ensure a smooth migration process.

Preparations

Before starting, please ensure you have the following:

1. A valid Azure subscription.
2. Administrative access to both the source and target regions.
3. Azure CLI installed and configured.
4. Before officially starting the regional migration, it is recommended to validate the migration scenarios and steps, and back up data.
5. Read the recommended guidelines in the Production Readiness Checklist.
6. Ensure there are no calls to the Service Fabric cluster resources, and no services are communicating with or executing processes.

Migration Steps

Create a New Service Fabric Cluster

Create a Service Fabric cluster using the Azure Portal or via Azure Resource Manager, see Creating Service Fabric clusters using Azure Resource Manager.

Set up a cluster in the new region by adjusting your existing ARM templates for the cluster and infrastructure topology, using your custom ARM template. If there is no ARM template describing the current cluster, it is recommended to retrieve the current ARM template from the Azure Resource Explorer. The Azure Resource Explorer helps you discover currently deployed resources and their configuration information, which you can use to create one or more ARM templates for repeatedly deploying clones of the existing environment. Ensure that you test and confirm you have a usable ARM template that can deploy a clone of the existing environment before proceeding.

Migrate Applications

1. **Deploy applications and services to the new Service Fabric cluster via Azure Resource Manager**, see Managing applications and services as Azure Resource Manager resources.

   Be sure to retain any application parameters or configuration customizations that were made. For example, if an application has a “count” parameter with a default value of 5, but this parameter was upgraded to 7 in the source environment, you must ensure that this value is set to 7 in the new region’s application deployment. If you are not using ARM to manage applications and service instances, you will be responsible for identifying the current applications and service sets running in the current region and replicating these applications and services in the new region/cluster.

2. Migrate Services

   – If it is a **stateful service**, data must be moved from the old cluster to the new cluster. Refer to Backup data from the old cluster to understand how to move data from the old cluster to the new cluster. For stateful workloads:

     • To ensure that stateful services have reached a stable point, make sure to stop incoming traffic to these services first. How to do this depends on different cases of how traffic is delivered to your services. For example, the service might need to be disconnected from the Event Hub, or traffic routing rules, like APIM or Azure Network Load Balancer rules, might need to be removed to prevent them from routing traffic to your service. Once traffic has stopped and the services have completed the background processing related to these requests, you can proceed.

     • Use the Backup Restore Service and perform an On-Demand Backup to back up any stateful services. Make sure to perform the backup after traffic has stopped and any background processing work has been completed. Once done, you can restore data to the stateful services in the new region and cluster. The Azure storage account used for backup must be accessible from the new region.

   – If it is a **stateless service**, no data needs to be moved, but traffic must be reconfigured. For stateless services:

     • Apart from deploying the services to the new cluster (preferably as part of the ARM application deployment done in step 2) and ensuring their configuration matches that in the source cluster, there should be no additional work required.

   – For all services:

     • Change the application configuration deployed in the new region.
     • Ensure any communication stages between clients and services are configured similarly to the source cluster. For instance, verifications might include ensuring that intermediary rules for Event Hubs, Network Load Balancers, Application Gateways, or API Management that allow traffic to flow to the clusters are set up.

Redirect Traffic

Redirect traffic from the old region to the new region. It is recommended to use Azure Traffic Manager for migration, as it provides various routing methods. How exactly to update traffic delivery rules depends on whether you are retaining or deprecating the existing region and how traffic is configured in your applications. You might need to investigate whether private/public IPs or DNS names can be moved between different Azure resources in different regions. Service Fabric does not recognize this part of the system, so please do the research and involve your Azure teams related to traffic flow if needed (especially if that part is very complex or delays would significantly impact your workload). Reviewing documents Configuring Custom Domains, Public IP Addresses, and DNS Zones and Records may be very helpful. Below are two example scenarios demonstrating how to update traffic routing:

- If you do not intend to retain the existing source region and have a DNS/CNAME associated with a public IP tied to a Network Load Balancer (which relays calls to the original source cluster). Update the DNS/CNAME to be associated with the new public IP of the new Network Load Balancer in the new region. Completing this transition will result in clients using the existing cluster switching to using the new cluster.

- If you intend to retain the existing source region and have a DNS/CNAME associated with a public IP tied to a Network Load Balancer (which relays calls to the original source cluster). Set up an instance of Azure Traffic Manager and then associate the DNS name with that Azure Traffic Manager instance. Then, Azure Traffic Manager can be configured to route to individual Network Load Balancers in each region.

If you intend to retain both regions, a kind of “reverse synchronization” usually takes place where the source of truth is retained in some remote storage (e.g., Azure SQL, Azure Cosmos DB, Blob, or File Storage) and synchronization between regions occurs. If this applies to your workload, it is recommended to confirm that data flows as expected between regions.

Validation and Testing

As a final validation, verify that traffic flows as expected and that the applications in the new region (and possibly still in the old region) run as expected. You can visualize this verification via Service Fabric Explorer, see Visualizing your cluster with Service Fabric Explorer.

1. Verify the status of the newly created Service Fabric cluster.

```
az sf cluster show `
--resource-group <NewResourceGroupName> `
--cluster-name <NewClusterName>
```

2. Check the cluster health to ensure all nodes are operational.

3. Verify that the applications deployed in the cluster are not missing.

```
az sf application list `
--resource-group <NewResourceGroupName> `
--cluster-name <NewClusterName>
```

4. Verify that the applications have been deployed to and are running on the Service Fabric cluster in the Azure target region.

5. Verify that there are no unforeseen applications accessing the applications and services in the Azure source region.

Clean Up Resources

If you do not plan to retain the original source region, you can delete the resources in that region at this time. It is recommended to wait some time before deleting the resources to ensure that any critical issue requiring a rollback to the source region can be addressed. See Delete the Resource Group Containing the Service Fabric Cluster.

Summary

By following the above steps, you have successfully migrated the Azure Service Fabric cluster from one region to another. Ensure to meticulously check each step during the migration process to avoid data loss or service interruption. If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

Learn how to migrate an Azure Service Fabric cluster. Relevant reference documents:

- Azure Service Fabric Overview
- Azure Service Fabric Documentation
- Deploy Windows containers to Service Fabric
- Deploy and remove applications using PowerShell

If you have any questions, please contact your Azure support team.

Migrate Azure Analysis Services

To migrate Azure Analysis Services across Azure regions, please refer to Move Analysis Services to a different region.

If you only want to migrate model metadata without the data, you can choose to Deploy a model from Visual Studio.

For more information: * Learn about Analysis Services backup and restore. * Review the Analysis Services overview.

If you have any questions, please contact your Azure support team.

Migrate Azure Backup Resource

Table of Contents

- Azure Backup Resource Region Migration Playbook
  - Table of Contents
  - Introduction
  - Prerequisites
  - Migration Steps
    - Stop Backup
    - Export Backup Data
    - Create Recovery Services Vault in Target Region
    - Import Backup Data
    - Reconfigure Backup Policies
  - Validation and Testing
  - Cleanup Resources
  - Summary

Introduction

This guide aims to instruct you on how to migrate Azure Backup resources from one region to another, ensuring data integrity and using Azure CLI for operations. This document specifically targets migration operations in the Azure in China regions. Please follow this playbook step-by-step to execute the migration tasks.

Prerequisites

Before you begin, make sure you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Azure CLI installed and configured.
4. Ensure the backup data is complete, and carefully read and back up related configuration.

Migration Steps

Stop Backup

First, stop your backup service in the source region to ensure no new backup data is generated.

```
# Use the name of the Recovery Services vault and its resource group
az backup protection disable --resource-group <SourceResourceGroup> --vault-name <SourceVaultName> --backup-management-type AzureIaasVM --item-name <BackupItemName> --workload-type VM --delete-backup-data false
```

Export Backup Data

Next, export the backup data to a storage account. In Azure Backup, the export operation can be accomplished by linking the backup and copying the snapshot to the storage account.

```
# Assuming a storage account has been created
export StorageAccountName=<YourStorageAccountName>
export StorageContainerName=<YourContainerName>
export ResourceGroupName=<ResourceGroup>
export BackupFileName=<BackupFileName>
```

```
az backup export_backup --resource-group $ResourceGroupName --vault-name <SourceVaultName> --containe
r-name $StorageContainerName --item-name <BackupItemName> --storage-account $StorageAccountName --
blob-container $StorageContainerName --retention-weekly 4 --retention-yearly 1
```

Ensure the backup export is successful. If there are any issues, rerun the above command.

Create Recovery Services Vault in Target Region

Create a new Recovery Services vault in the target region to receive the imported data.

```
az backup vault create --resource-group <TargetResourceGroup> --name <TargetVaultName> --location <Targe
tRegion>
```

Import Backup Data

Move the backup data from the source storage account to the target storage account and register it in the new Recovery Services vault.

```
# Assuming new storage account and container are created
export TargetStorageAccountName=<TargetStorageAccountName>
export TargetStorageContainerName=<TargetContainerName>
```

```
az storage blob copy start-batch --destination-container $TargetStorageContainerName --account-name $Target
StorageAccountName --source-container $StorageContainerName --source-account-name $StorageAccountNam
e --destination-share <NewBackupsFileShare>
```

Then, register the backup data in the target region's new Recovery Services vault.

```
az backup container register --vault-name <TargetVaultName> --backup-management-type AzureIaasVM --reso
urce-group <TargetResourceGroup> --location <TargetRegion>
```

Reconfigure Backup Policies

Reconfigure the backup policies in the target Recovery Services vault to ensure the new configuration matches the previous one.

```
az backup policy create --name <PolicyName> --resource-group <TargetResourceGroup> --vault-name <Target
VaultName> --schedule-policy <NewSchedulePolicy> --retention-policy <NewRetentionPolicy> --backup-manag
ement-type AzureIaasVM --workload-type VM
```

and respectively represent the new backup schedule and retention policies.

Validation and Testing

1. Start the backup service in the target region and verify its status.

```
az backup protection enable-for-vm --vault-name <TargetVaultName> --resource-group <TargetResourceGroup
> --vm <VMName> --policy-name <PolicyName>
```

2. Check if the backup data is complete and conduct partial data recovery tests for verification.

```
az backup recoverypoint show --resource-group <TargetResourceGroup> --vault-name <TargetVaultName> --c
ontainer-name <ContainerName> --item-name <BackupItemName> --recovery-point-id <RecoveryPointId>
```

Cleanup Resources

If the new backup service is running smoothly, delete the old backup service and related resources in the source region.

```
az backup vault delete --resource-group <SourceResourceGroup> --name <SourceVaultName> --yes
```

Summary

By following the above steps, you have successfully migrated Azure Backup from one region to another. Ensure you carefully check each step during the migration to avoid data loss or service interruptions. If you encounter any problems, refer to the official Azure documentation or contact Azure technical support.

Migrate Azure Cache for Redis Instance

This article describes how to move an Azure Cache for Redis instance to a different Azure region.

- If the current cache tier is Premium, you can use: Geo-replication, creating a new cache, performing dual writes to both caches, exporting and importing data via RDB files, or migrating programmatically.
- If the current cache tier is Basic or Standard, you can use: creating a new cache, performing dual writes to both caches, or migrating programmatically.

Passive Geo-Replication (Premium)

Prerequisites

To configure Geo-replication between two caches, the following prerequisites must be met:

- Both caches are Premium tier caches.
- Both caches are in the same Azure subscription.
- The secondary link cache size is equal to or larger than the primary link cache size.
- Both caches already exist and are running.

Preparation

To move a cache instance to another region, you need to create a second Premium cache instance in the desired region. Once both caches are running, you can set up Geo-replication between them.

Some features are not supported with Geo-replication:

- Geo-replication does not support zone redundancy.
- Geo-replication does not support persistence.

Conditions where geo-replication is supported:

- Cluster is supported if both caches have clustering enabled and have the same number of shards.
- Caches in different VNets are also supported, but some issues need to be noted. For more information, see Can I use geo-replication with my caches in a VNet?.

After the Geo-replication configuration is completed, the linked cache pair will have the following restrictions:

- The secondary linked cache is read-only. You can read data from it, but you cannot write any data to it.
  - If you choose to read data from the geo-secondary instance while synchronizing it with the geo-primary instance (e.g., when either instance is updating or in certain restart scenarios), the geo-secondary instance will raise errors for any Redis operations against it until the synchronization is complete between the geo-primary and geo-secondary.
  - You should design applications reading from the geo-secondary to fallback to the geo-primary when such errors occur.
- Any data in the secondary linked cache before setting up the link will be deleted. However, if you remove geo-replication later, the replicated data will remain in the secondary linked cache.
- You cannot scale either cache while they are linked.
- You cannot change the shard count if clustering is enabled.
- You cannot enable staging on either cache.
- You can export from either cache.
- You cannot import into the secondary linked cache.

- Neither the linked caches nor their resource groups can be deleted unless the caches are unlinked first. For more details, see Why did the operation fail when I tried to delete my linked cache?.
- Network egress charges apply for moving data between regions if the caches are in different regions. For more details, see How much does it cost to replicate my data across Azure regions?.
- Failover does not happen automatically. You must initiate failover from the primary link cache to the secondary link cache. For details on how to failover client applications, see Initiate a failover from geo-primary to geo-secondary.

Move

1. To link two caches for Geo-replication, first select “Geo-Replication” in the “Resource” menu of the cache you want to use as the primary link cache. Then, in “Geo-Replication” on the left, select “Add Cache Replication Link”.

2. In the “Compatible caches” list, select the name of the desired secondary cache. If the secondary cache is not displayed in the list, confirm whether it meets the Geo-replication prerequisites for the secondary cache. To filter caches by region, select the respective region in the map to show only caches in the “Compatible caches” list. You can also use the context menu to start the link process or view details of the secondary cache.

3. Select “Link” to link the two caches and start the replication process.

   For more information, see Move Azure Cache for Redis instance to different region.

Verify

You can use “Geo-Replication” on the left to view the progress of the replication process. You can also use the “Overview” to view the link status of the primary and secondary caches on the left. Once the replication process is completed, the “Link Status” changes to “Succeeded”. During the link process, the primary link cache remains available. The secondary link cache will be unavailable until the link process is complete.

Create a New Cache (All Tiers)

Preparation

If maintaining data during the move is not required, the simplest way to move regions is to create a new cache instance in the target region and connect your application to this instance. For example, if you use Redis as a fallback cache for database records, you can easily rebuild the cache from scratch.

For specific steps, see Move Azure Cache for Redis instance to different region.

Note: Creating the cache takes some time. You can monitor the progress on the “Overview” page of Azure Cache for Redis. If the “Status” shows “Running,” the cache is available for use.

Finally, update the application to use the new instance.

Export and Import Data via RDB Files (Premium)

Open source Redis defines a standard mechanism to take a snapshot of the in-memory dataset of a cache and save it to a file. This file is called an RDB and can be read by another Redis cache. Azure Cache for Redis Premium supports importing data into cache instances via RDB files. You can use RDB files to transfer data from an existing cache to Azure Cache for Redis.

Note: The RDB file format may change between Redis versions and may not maintain backward compatibility. The Redis version of the cache to be exported should not be higher than that of the new cache instance.

Prerequisites

- Both caches are Premium tier caches.
- The size of the second cache should be equal to or larger than the original cache.
- The Redis version of the cache to be exported should not be higher than that of the new cache instance.

Preparation

To move a cache instance to another region, you need to create a second Premium cache instance in the desired region.

Move

1. For more information on how to import and export data in Azure Cache for Redis, see Import and Export Data in Azure Cache for Redis.

2. Update the application to use the new cache instance.

Verify

You can monitor the progress of the import operation through the notifications in the Azure portal or by examining events in the audit logs.

Dual Writes to Both Caches (Basic, Standard, and Premium)

Instead of moving data directly between caches, you can have your application write data to both the existing cache and the new set cache. Initially, the application reads data from the existing cache. Once the new cache has the necessary data, you can switch the application to use the new cache and decommission the old cache. For example, assuming Redis is used as session storage and the application's sessions last 7 days, after a week of dual writes, the new cache will contain all unexpired session information. After this, you can safely rely on it without worrying about data loss.

Prerequisites

- The size of the second cache should be equal to or larger than the original cache.

Preparation

To move a cache instance to another region, you need to create a second cache instance in the desired region.

Move

The general steps to implement this option are as follows:

1. Modify application code to write to both the new instance and the original instance.

2. Continue to read data from the original instance until the new instance is sufficiently populated.

3. Update the application code to read and write solely from the new instance.

Migrate Programmatically (All Tiers)

You can create a custom migration process by programmatically reading data from the existing cache and writing it to Azure Cache for Redis. This open source tool can be used to copy data from one Azure Cache for Redis instance to another instance in a different Azure cache region. A compiled version is also available. You may also find the source code provides useful guidance for writing your own migration tool.

Prerequisites

- The size of the second cache should be equal to or larger than the original cache.

Preparation

- Create a VM in the region of the existing cache. If the dataset is large, select a relatively powerful VM to reduce replication time.
- To move a cache instance to another region, you will need to create a second cache instance in the desired region.

Move

After creating a VM in the region of the existing cache and creating a new cache in the desired region, the general steps to implement this option are:

1. Flush data from the new cache to ensure it's empty. This step is necessary because the copy tool itself will not overwrite any existing keys in the target cache.
2. Use a tool (such as the aforementioned open source tool) to automatically copy data from the source cache to the target cache. Keep in mind that the replication process might take some time to complete, depending on the size of the dataset.

Migrate Azure Database for MySQL

Single Server

Azure Database for MySQL Single Server is scheduled to be retired by September 16, 2024. It is recommended to migrate Azure Database for MySQL Single Server to Azure Database for MySQL Flexible Server.

Prerequisite Checks and Post-Migration Actions When Migrating from Single Server to Flexible Server

- If the source Azure Database for MySQL Single Server has engine version v8.x, ensure to upgrade the .NET client driver version of the source server to 8.0.32 to avoid any coding compatibility issues after migration to the Flexible Server.
- If the source Azure Database for MySQL Single Server has engine version v8.x, ensure to upgrade the TLS version of the source server from v1.0 or v1.1 to TLS v1.2 before migration since older TLS versions are deprecated on Flexible Server.
- If the source Azure Database for MySQL Single Server uses non-default ports (such as 3308, 3309, and 3310), change the connection port to 3306 since Flexible Server does not support the aforementioned non-default ports.
- Service tags (SQL) in outbound rules are not supported on Azure Database for MySQL Flexible Server. When configuring firewall settings for the Flexible Server, use fully qualified domain names (FQDN) in outbound rules.

Migrating from Single Server to Flexible Server

Learn how to migrate from Azure Database for MySQL Single Server to Azure Database for MySQL Flexible Server.

Choose the appropriate tool based on your actual scenario:

Scenario	Tool	Details
Offline/Online	Azure Database for MySQL Import and Azure CLI	Tutorial: Azure Database for MySQL Import Using Azure CLI
Offline	Database Migration Service (Classic) and Azure Portal	Tutorial: Azure Portal and DMS (Offline)
Online	Database Migration Service (Classic) and Azure Portal	Tutorial: Azure Portal and DMS (Online)

To learn more about using other migration tools to migrate from Single Server to Flexible Server, visit Choosing the Right Tool for Migration to Azure Database for MySQL.

Related Content

- Azure Database for MySQL - What's Happening to Single Server
- Azure Database for MySQL - Azure Regions for Flexible Server

Migrate Azure Database For PostgreSQL

Azure Database for PostgreSQL - The Single Server plan will be retired by March 28, 2025. It is recommended to migrate Azure Database for PostgreSQL - Single Server to Azure Database for PostgreSQL - Flexible Server.

Migrate from Azure Database for PostgreSQL - Single Server to Azure Database for PostgreSQL - Flexible Server

Learn how to use the Single Server to Flexible Server Migration Tool to migrate from Azure Database for PostgreSQL - Single Server to Azure Database for PostgreSQL - Flexible Server.

Related Content

- Latest Updates on Azure Database for PostgreSQL Single Server
- Azure Database for PostgreSQL - Flexible Server Azure Regions

Migrate Azure SQL Resource

To move Azure SQL resources from one region to another, you can use bacpac files for export and import.

Considerations

- To ensure transactional consistency during export, you must ensure no write activity occurs during the export period or you are exporting from a transactionally consistent copy of the database.
- If exporting to Blob storage, the maximum size for a BACPAC file is 200 GB. To archive larger BACPAC files, use SqlPackage to export to local storage.
- Azure storage file names must not end with a . and cannot contain space characters or special characters like <, >, *, %, &, :, \, /, ?, etc. The file name length should be less than 128 characters.
- If the export operation exceeds 20 hours, the operation might be canceled. To improve performance during the export process, you can:
  - Temporarily increase compute size.
  - Terminate all read and write activities during the export.
  - Use a clustered index on all large tables with non-null values. If you do not use a clustered index, the export may fail after 6-12 hours because the export service needs to complete a table scan to attempt exporting the entire table. A good way to confirm whether the table is optimized for export is to run DBCC SHOW_STATISTICS and ensure that *RANGE_HI_KEY* is not null and the value distribution is good. For more details, refer to DBCC SHOW_STATISTICS.
- For larger databases, BACPAC export/import might take a long time and could fail for various reasons.

BACPAC File Export

Note: Currently, exporting a BACPAC file for a database from an Azure SQL Managed Instance is not supported using Azure Portal. Refer to considerations.

Refer to Export a database to a BACPAC file.

BACPAC File Import

You can import an SQL Server database into Azure SQL Database or SQL Managed Instance using a .bacpac file. Data can be imported from a .bacpac file stored in Azure Blob Storage (standard storage only) or from local storage at a local location. To maximize import speed by providing more and faster resources, scale up the database to a higher service tier and a larger compute size during the import process. Then, you can scale down after a successful import.

Note: Bacpac files generated from SqlPackage that exceed 4GB might fail to import from the Azure Portal or Azure PowerShell displaying an error message stating File contains corrupted data. This is a known issue, and the workaround is to use the SqlPackage command-line utility to import the bacpac file. For more details, refer to SqlPackage and the issue log.

Note: To migrate a database from a bacpac file to an Azure SQL Managed Instance, use SQL Server Management Studio or SQLPackage. Using Azure Portal or Azure PowerShell is not currently supported.

Refer to Import a database from a bacpac file.

Verification and Testing

1. **Verify Database Connection**: Verify if the database connection is working correctly by querying data in SQL.

2. **Verify Applications**: Start and run applications or services dependent on Azure SQL to ensure they are working properly.

Migrate SQL Server Stretch Database

Stretch Database is deprecated in SQL Server 2022 (16.x) and Azure SQL Database. This feature will be removed in a future version of the Database Engine. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Related Content

- Stretch Database - SQL Server Stretch Database | Microsoft Learn

Migrate Azure Monitor

Table of Contents

- Overview
- Log Analytics Workspace
  - Prerequisites
  - Downtime
  - Migration Steps
- Application Insights
  - Availability Tests
- Workbook
- Summary

Overview

Azure Monitor feature is non-regional and no migration is required.
However, some features included in Azure Monitor may be regional, such as Log Analytics Workspace, Application Insights, Availability Tests, Workbooks, etc.
The regions supported by different features may vary slightly and may change with product iterations and upgrades.
It is important to confirm early during the migration planning phase whether the features are supported in the target region and plan accordingly.

Log Analytics Workspace

A relocation plan for Log Analytics workspace must include the relocation of any resources that log data with Log Analytics Workspace.
Log Analytics workspace doesn't natively support migrating workspace data from one region to another and associated devices.
Instead, you must create a new Log Analytics workspace in the target region and reconfigure the devices and settings in the new workspace.

Prerequisites

- To export the workspace configuration to a template that can be deployed to another region, you need the Log Analytics Contributor or Monitoring Contributor role, or higher.

- Identify all the resources that are currently associated with your workspace, including: connected agents, diagnostic settings, installed solutions, data collector API, linked services, alerts, query packs. (Reference)
- Verify that your Azure subscription allows you to create Log Analytics workspaces in the target region.

Downtime

To understand the possible downtimes involved, see Cloud Adoption Framework for Azure: Select a relocation method.

Migration Steps

You need to first export the template of the source Log Analytics Workspace, and after making adjustments, redeploy it in the new target region.
For detailed migration steps, refer to Relocate Azure Monitor - Log Analytics workspace to another region.

Application Insights

Transferring existing Application Insights resources between regions isn't supported, and you can't migrate historical data to a new region.
For workaround involves, refer to How do I move an Application Insights resource to a new region.

Availability Tests

The availability tests feature relies on workspace-based Application Insights resources, but the execution region of the availability tests is independent of the region where the Application Insights resource is located.
You can create or edit availability tests in different regions in the Availability pane of the Application Insights resource.
For more details, refer to Application Insights Availability Tests.

Workbook

If you want to move Azure Workbook resources to another Azure region, refer to Move an Azure Workbook to another region.

Summary

Before migrating in a production environment, test and validate in a test environment.
If you encounter any issues, refer to Azure official documentation or contact Azure technical support.

For more information:

- Azure Monitor
- Log Analytics workspace overview
- Application Insights overview

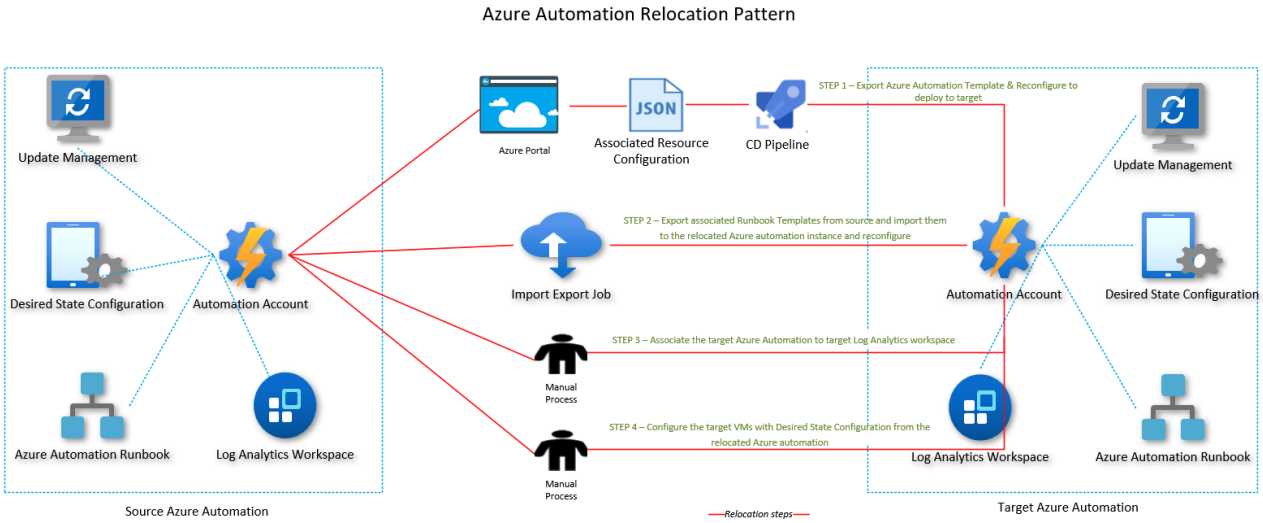
Migrate Azure Automation

Overview

Azure Automation delivers a cloud-based automation, operating system updates, and configuration service that supports consistent management across your Azure and non-Azure environments. It includes process automation, configuration management, update management, shared capabilities, and heterogeneous features.

Azure Automation cannot directly move from one region to another. One solution is to “export the template” from the old region, modify the parameters in the new region, recreate Azure Automation, and associate the moved resources with the new instance. Once the new Azure Automation is running in the new region, you can delete the old automation in the previous region.

In the diagram below, the red streamlined flow demonstrates the redeployment of the target instance and configuration move. For more information, please read Relocate Azure Automation to another region.



Azure Automation Relocation Pattern

Related Reference Documentation

- Azure Automation Documentation

Migrate Azure Firewall

Overview

Azure Firewall is a cloud-native network security service that offers high availability and seamless cloud scaling. This guide will help you migrate Azure Firewall from one region to another.

Prerequisites

- It is strongly recommended to use Premium SKU. If you are using the Standard SKU, consider Migrate Azure Firewall configurations to Azure Firewall policy using PowerShell before relocation.
- To properly plan and execute the Azure Firewall relocation, you must gather the following information:
  - **Deployment Model**: Classic Firewall Rules or Firewall Policy.
  - **Firewall Policy Name**: (If using the Firewall Policy deployment model).
  - **Diagnostic Settings at the Firewall Instance Level**: (If using Log Analytics workspace).
  - TLS (Transport Layer Security) Inspection Configurations: (If using Azure Key Vault, certificates, and managed identities).
  - **Public IP Control**: Assess whether any external identities depending on Azure Firewall public IP remain fixed and trusted.
- Azure Firewall Standard and Premium tiers have the following dependencies that you may need to deploy in the target region:
  - Azure Virtual Network
  - (If used) Log Analytics
- If using the TLS inspection feature of Azure Firewall Premium, you will also need to deploy the following dependencies in the target region:
  - Azure Key Vault
  - Azure Managed Identities

Downtime

To understand the potential downtime involved, refer to Cloud Adoption Framework for Azure: Select a Relocation Method.

Migration Steps

Step 1: Create Resources in the Target Region

1. Create a new Virtual Network (VNet) in the target region.
2. Configure the required subnets, including AzureFirewallSubnet, in the new VNet.

Step 2: Create a New Azure Firewall

1. [Optional] If the original Azure Firewall uses classic rules, you can migrate the existing classic rules in Azure Firewall to create policies using Azure PowerShell. For more details, see migrate Azure Firewall configuration to Azure Firewall policies using PowerShell.
2. If already using Azure Firewall policies, you can migrate existing policies using Azure PowerShell or save and redeploy policies by selecting the desired Azure Firewall policy and clicking **Export Template** through the Azure Portal.

Redeploy-Policy.ps1 is an Azure PowerShell script to redeploy an existing policy to a new region.

Usage example:

```
.\Redeploy-Policy.ps1 `
  -PolicyId <ResourceId> `
  -TargetResourceGroup target-resource-group `
  -NewPolicyName new-policy-name `
  -Location chinanorth3 `
  -SkuTier Standard
```

Important The script does not migrate threat intelligence and SNAT private range settings. You will need to note these settings first and migrate them manually. This script is provided as an example and is not intended for direct migration use in production environments.

This script requires the latest Azure PowerShell. Run Get-Module -ListAvailable Az to see which versions are installed. If installation is needed, refer to Install Azure PowerShell modules.

```
param (
  # Resource ID of the original Azure firewall policy.
  [Parameter(Mandatory=$true)]
  [string]
  $PolicyId,

  # Target resource group for the new policy.
  [Parameter(Mandatory=$true)]
  [string]
  $TargetResourceGroup,

  # New name for the Azure firewall policy.
  [Parameter(Mandatory=$true)]
  [string]
  $NewPolicyName,

  # Location for the new Azure firewall policy.
  [Parameter(Mandatory=$true)]
  [string]
  $Location,

  # SKU Tier for the new Azure firewall policy.
  [Parameter(Mandatory=$false)]
  [string]
  $SkuTier = "Premium"
)

$ErrorActionPreference = "Stop"

function EnsureAzureChinaLogin {
  Write-Host "Checking Azure login status"
  try {
    $account = Get-AzContext
```

```
        if ($null -eq $account) {
            throw "Not logged in"
        }
        Write-Host "Already logged in as: $($account.Account)"
    }
    catch {
        Write-Host "Not logged in. Logging in to Azure China..."
        Connect-AzAccount -Environment AzureChinaCloud
        $account = Get-AzContext
        if ($null -eq $account) {
            Write-Host "Login failed. Please check your credentials and try again."
            exit(1)
        }
        Write-Host "Logged in successfully as: $($account.Account)"
    }
}

function CreateNewPolicy {
    [CmdletBinding()]
    param (
        [Parameter(Mandatory=$true)]
        [Microsoft.Azure.Commands.Network.Models.PSAzureFirewallPolicy]
        $OriginalPolicy,
        [Parameter(Mandatory=$true)]
        [string]
        $NewPolicyName,
        [Parameter(Mandatory=$true)]
        [string]
        $TargetResourceGroup,
        [Parameter(Mandatory=$true)]
        [string]
        $Location,
        [Parameter(Mandatory=$true)]
        [string]
        $SkuTier
    )

    $NewPolicyParameters = @{
        Name = $NewPolicyName
        ResourceGroupName = $TargetResourceGroup
        Location = $Location
        ThreatIntelMode = $OriginalPolicy.ThreatIntelMode
        ThreatIntelWhitelist = $OriginalPolicy.ThreatIntelWhitelist
        PrivateRange = $OriginalPolicy.PrivateRange
        DnsSetting = $OriginalPolicy.DnsSettings
        SqlSetting = $OriginalPolicy.SqlSetting
        ExplicitProxy  = $OriginalPolicy.ExplicitProxy
        DefaultProfile  = $OriginalPolicy.DefaultProfile
```

```
        Tag = $OriginalPolicy.Tag
        SkuTier = $SkuTier
    }

    Write-Host "Creating new policy"
    $newPolicy = New-AzFirewallPolicy @NewPolicyParameters

    Write-Host "Populating rules in new policy"
    foreach ($ruleCollectionGroup in $OriginalPolicy.RuleCollectionGroups) {
        $ruleResource = Get-AzResource -ResourceId $ruleCollectionGroup.Id
        $ruleToTransfom = Get-AzFirewallPolicyRuleCollectionGroup -AzureFirewallPolicy $OriginalPolicy -Name $ruleResource.Name
        $ruleCollectionGroup = @{
            FirewallPolicyObject = $newPolicy
            Priority = $ruleToTransfom.Properties.Priority
            Name = $ruleToTransfom.Name
        }

        if ($ruleToTransfom.Properties.RuleCollection.Count) {
            $ruleCollectionGroup["RuleCollection"] = $ruleToTransfom.Properties.RuleCollection
        }

        Set-AzFirewallPolicyRuleCollectionGroup @ruleCollectionGroup
    }

    return $newPolicy
}

function ValidateAzNetworkModuleExists {
    Write-Host "Validating needed module exists"
    $networkModule = Get-InstalledModule -Name "Az.Network" -MinimumVersion 4.5 -ErrorAction SilentlyContinue
    if ($null -eq $networkModule) {
        Write-Host "Please install Az.Network module version 4.5.0 or higher, see instructions: https://github.com/Azure/azure-powershell#installation"
        exit(1)
    }
    $resourceModule = Get-InstalledModule -Name "Az.Resources" -MinimumVersion 4.2 -ErrorAction SilentlyContinue
    if ($null -eq $resourceModule) {
        Write-Host "Please install Az.Resources module version 4.2.0 or higher, see instructions: https://github.com/Azure/azure-powershell#installation"
        exit(1)
    }
    Import-Module Az.Network -MinimumVersion 4.5.0
    Import-Module Az.Resources -MinimumVersion 4.2.0
}
```

```
EnsureAzureChinaLogin
ValidateAzNetworkModuleExists
$originalPolicy = Get-AzFirewallPolicy -ResourceId $PolicyId
$newPolicy = CreateNewPolicy -OriginalPolicy $originalPolicy -NewPolicyName $NewPolicyName -TargetResourc
eGroup $TargetResourceGroup -Location $Location -SkuTier $SkuTier

Write-Host "Migration complete. New policy created: $($newPolicy.Name)" -ForegroundColor Green
```

3. Create a new Azure Firewall instance in the target region and select the migrated Azure Firewall policy.

Step 3: Test and Validate

1. Ensure the new Azure Firewall instance is successfully created and configured.
2. Conduct connectivity tests to verify that firewall rules are properly applied.
3. Confirm that logging and monitoring settings are functioning correctly.

Step 4: Migrate Traffic

1. Update relevant routing tables and Network Security Groups (NSGs) to point to the new Azure Firewall instance.
2. Gradually switch the traffic to the new firewall to ensure no interruptions.

Step 5: Clean Up Old Resources

1. Once confirmed that the new Azure Firewall is working correctly, you can delete the old Azure Firewall instance.
2. Clean up resources no longer in use in the old region to save costs.

Considerations

- Ensure the migration is performed during off-peak hours to minimize business impact.
- Closely monitor network traffic and logs during the migration to ensure there are no issues.
- If any issues arise, you can roll back to the original Azure Firewall instance at any time.

Related References

- Azure Firewall Documentation
- Azure CLI Documentation
- Azure Network Security Group (NSG) Documentation
- Azure Routing Table Documentation

By following these steps, you can smoothly migrate Azure Firewall from one region to another. For any issues, refer to the aforementioned documents or contact the Azure support team.

Migrate Azure Key Vault

Table of Contents

- Introduction
- Prerequisites
- Considerations
  - Service Endpoints Considerations
  - Private Endpoints Considerations
  - Azure Private Endpoint DNS Integration Considerations
- Preparations
- Migration Steps
  - Stop Applications
  - Backup Key Vault
  - Create Key Vault in Target Region
  - Restore Key Vault Data
  - Update Application Configuration
  - Validation and Testing
  - Clean Up Resources
- Summary

Introduction

This manual aims to guide how to migrate Azure Key Vault from one region to another, including operations between China regions. We will use Azure CLI tools and detail each step to ensure data integrity and minimal service disruption.

Prerequisites

- Verify that your Azure subscription allows the creation of a Key Vault in the target region.
- Create a dependency map for all Azure services used by the Key Vault. For services within the relocation scope, choose appropriate relocation strategies.
- Depending on the Key Vault design, you may need to deploy and configure virtual networks in the target region.
- To create keys, secrets, and certificates, add users to the “Key Vault Contributor” role.
- Document and plan the reconfiguration of the Key Vault in the target region:
  - Access policies and network configuration settings.
  - Soft delete and purge protection.
  - Auto-rotation settings.

Considerations

Service Endpoints Considerations

Virtual network service endpoints via Azure Key Vault can restrict access to specified virtual networks. Additionally, these endpoints can limit access to a range of IPv4 (Internet Protocol version 4) address ranges.

Any external users connecting to the Key Vault will not be able to access these resources. If service endpoints were configured in the source region for the Key Vault resources, you must perform the same configuration in the target region.

To successfully recreate the Key Vault in the target region, create VNet and subnets beforehand, then proceed to the actual recreation.

Private Endpoints Considerations

Azure Private Link provides a private connection from a virtual network to Azure platform as a service (PaaS), customer-owned services, or Microsoft partner services. Private Link simplifies the network architecture by eliminating data exposure over the public Internet to protect connections between endpoints in Azure.

To successfully recreate the Key Vault in the target region, create VNet and subnets beforehand, then proceed to the actual recreation.

Azure Private Endpoint DNS Integration Considerations

Ensure proper DNS configuration so that private endpoint IP addresses resolve to the fully qualified domain name (FQDN) of the connection strings.

Existing Microsoft Azure services might have DNS configurations for public endpoints. This configuration must be overridden to connect using a private endpoint.

The network interface associated with the private endpoint contains the information needed for DNS configuration. The network interface information includes the FQDN and private IP address of the Private Link resource.

Use the following options to configure DNS settings for the private endpoint:

- **Use host files (recommended only for testing).** You can use host files on virtual machines to override DNS.
- **Use private DNS zones.** Use private DNS zones to override DNS resolution of private endpoints. You can link private DNS zones to your virtual network to resolve specific domains.
- **Use DNS forwarders (optional).** Use DNS forwarders to override DNS resolution for Private Link resources. Create DNS forwarding rules to use private DNS zones on DNS servers hosted within your virtual network.

Preparations

Before starting, ensure you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Azure CLI installed and configured.
4. Ensure your Key Vault has no ongoing operations and that data is backed up.

Migration Steps

Stop Applications

First, stop all applications using the Key Vault to ensure that Key Vault data does not change during the migration process.

Backup Key Vault

1. Log in to the Azure portal.
2. Navigate to the Key Vault you want to migrate.
3. In the left navigation pane under the “Objects” section, select “Keys.”
4. Click on the row of the key you want to back up in the right-side list.
5. On the key details page, click “Download Backup” and save the backup file.
6. Follow the steps above to back up “Secrets” and “Certificates.”

Use Azure CLI to back up all keys, secrets, and certificates in the existing Key Vault.

```
# 备份所有密钥
az keyvault key backup `
--vault-name <SourceKeyVaultName> `
--name <KeyName> `
--file <KeyBackupFilePath>

# 备份所有机密
az keyvault secret backup `
--vault-name <SourceKeyVaultName> `
--name <SecretName> `
--file <SecretBackupFilePath>

# 备份所有证书
az keyvault certificate backup `
--vault-name <SourceKeyVaultName> `
--name <CertificateName> `
--file <CertificateBackupFilePath>
```

Create Key Vault in Target Region

1. In the Azure portal, navigate to “Create a resource”.
2. Search for and select “Key Vault”.
3. Fill in the basic information to create the Key Vault and select the target region (i.e., the migration destination region).
4. Complete the creation and wait for the resource to deploy.
5. Configure Key Vault using Role-Based Access Control (RBAC) or Access Policies.

Use Azure CLI to create a new Key Vault in the target region.

```
# Create Key Vault
az keyvault create `
--name <TargetKeyVaultName> `
--resource-group <TargetResourceGroupName> `
--location <TargetRegion>
```

Restore Key Vault Data

1. Navigate to the newly created Key Vault.
2. In the left navigation pane under the “Objects” section, select “Keys”.
3. Click the “Restore Backup” button at the top right.

4. Select the corresponding backup file and wait for the system to restore the keys.
5. Follow the same steps to restore “Secrets” and “Certificates”.

Use Azure CLI to restore the backed-up data to the new Key Vault in the target region.

```
# Restore all keys
az keyvault key restore `
--vault-name <TargetKeyVaultName> `
--file <KeyBackupFilePath>
# Restore all secrets
az keyvault secret restore `
--vault-name <TargetKeyVaultName> `
--file <SecretBackupFilePath>

# Restore all certificates
az keyvault certificate restore `
--vault-name <TargetKeyVaultName> `
--file <CertificateBackupFilePath>
```

Update Application Configuration

Update application configurations to point to the new Key Vault and restart these applications.

```
# Update application's connection strings or configuration files to point to the new Key Vault
# How to update this depends on the type of your application
```

Validation and Testing

1. Restart the applications and verify they can normally access the new Key Vault.

2. Ensure all keys, secrets, and certificates can be read and used correctly.

```
# For example, verify if the application can correctly read the secrets
```

Clean Up Resources

If the new Key Vault and applications are confirmed to be running normally, you can delete the old Key Vault and related resources in the source region.

```
az keyvault delete `
--name <SourceKeyVaultName> `
--resource-group <SourceResourceGroupName>
```

Summary

By following the above steps, you have successfully migrated Azure Key Vault from one region to another. Ensure to double-check each step during the migration process to avoid data loss or service interruption. If you encounter any issues, refer to the official Azure documentation or contact Azure technical support.

Learn more about Azure Key Vault operations and reference documents:

- Azure Key Vault Tutorial

- Overview of Azure Key Vault

For any questions, please contact your Azure support team.

Migrate Azure Managed Disks

Table of Contents

- Introduction
- Migration Steps
  - Step 1: Get the shared access signature URI
  - Step 2: Copy to storage account
    - Step 2.1: AzCopy
    - Step 2.2: Azure CLI
  - Step 3: Create a new managed disk in the target environment
  - Step 4: Create the VM
- Summary

Introduction

Azure Managed Disks simplifies disk management for Azure infrastructure as a service (IaaS) VMs by managing the storage accounts that are associated with the VM disk.
Because you don't have direct access to the .vhd file, you can't directly use tools like AzCopy to copy your files. The workaround provided in this document involves exporting the managed disk first by obtaining a temporary shared access signature URI, then downloading or copying it using this URI, and finally creating a new managed disk in the target region based on the .vhd file.

Migration Steps

Step 1: Get the shared access signature URI

1. In the portal, search for your managed disk. It's in the same resource group as your VM and has a resource type of Disk.
2. On the Overview page, select the Export button in the top menu. You must shut down and deallocate your VM first, or unattach the VM to complete the export.
3. Define a time for the URI to expire. The default time is 3,600 seconds.
4. Generate a URL.
5. Copy the URL. The url will only be showed one time after creation.

Step 2: Copy to storage account

There are multiple ways to copy the exported disk to the target storage account.
This document uses AzCopy and Azure CLI as examples. You can choose to use either method.

Step 2.1: AzCopy

Use AzCopy to copy the disk directly from your source environment to the storage account blob container of the target environment.

For details on how to use AzCopy, please refer to Get started with AzCopy.

The AzCopy command is as follows:

```
azcopy copy "<source SAS URI>" "<target SAS URI>" --blob-type PageBlob
```

Here's a complete command example:

```
azcopy copy
 "https://md-fvth500tz1jn.blob.core.chinacloudapi.cn/nd2vhh3qsbn2/abcd?sv=2018-03-28&sr=b&si=08af14df-4aea-444c-89de-06c251f213ae&sig=xxx"
 "https://targetstorage.blob.core.chinacloudapi.cn/targetcontainer/newdisk.vhd?sp=rcw&st=2024-09-13T03:16:24Z&se=2024-09-13T11:16:24Z&spr=https&sv=2022-11-02&sr=c&sig=xxx"
 --blob-type PageBlob
```

Important Notes:
- Ensure the validity and permission settings of the URI are correct, especially when dealing with SAS tokens. - Use –blob-type PageBlob since managed disks are usually stored as page blobs. - Check the network settings, such as firewall and virtual network configurations, of the source and target storage accounts to ensure AzCopy can successfully connect to both.

Step 2.2: Azure CLI

Use Azure CLI to copy the disk directly from your source environment to the storage account blob container of the target environment.

The Azure CLI command is as follows:

```
az storage blob copy start
 --destination-blob "<the target VHD file name>"
 --destination-container "<the target storage account container name>"
 --account-name "<the target storage account name>"
 --account-key "<the target storage account key>"
 --source-uri "<source SAS URI>"
```

Of course, you can also complete the process in Step 1: Get the shared access signature URI through Azure CLI. This is undoubtedly a good way if you want to automate the migration of a large number of managed disks through scripts.

For the complete process and script, see Export/Copy a managed disk to a storage account using the Azure CLI.

Step 3: Create a new managed disk in the target environment

There are several options for creating a new managed disk. Here's how to do it in the Azure portal:

1. In the portal, select **New > Managed Disk > Create**.
2. Enter a name for the new disk.
3. Select a resource group.
4. Under **Source Type**, select **Storage Blob**. Then, either copy the destination URI from the AzCopy command or browse to select the destination URI.
5. If you copied an OS disk, select the **OS** type. For other disk types, select **Create**.

Step 4: Create the VM

As mentioned, there are multiple ways to create a VM by using this new managed disk. Here are two options: * In the portal, select the disk, and then select **Create VM**. Define the other parameters of your VM as usual. * For PowerShell, see Create a VM from restored disks.

Summary

By following the above steps, you have successfully migrated an Azure managed disk from one region to another.

Make sure to double-check each step during the migration to avoid data loss or service interruption.

Before migrating in a production environment, test and validate in a test environment.

If you encounter any issues, refer to Azure official documentation or contact Azure technical support.

To learn more about managing Azure managed disks, refer to:

- Azure Managed Disks
- Learn how to export to disk via API by getting a shared access signature URI.
- Learn how to create a managed disk via API from an unmanaged blob.

Migrate Azure Storage Account

Overview

This section will guide you on how to migrate an Azure Storage Account from one region to another. Since Azure does not support directly changing the region of a storage account, migration must be done through backup and data replication.

Prerequisites

Before starting the migration, ensure the following conditions are met:

- Ensure the services and features used by the account are supported in the target region.
- For preview features, ensure your subscription is whitelisted for the target region.
- Depending on the deployment of the storage account, you may need to deploy and configure the following dependent resources in the target region before relocation:
  - Virtual Networks, Network Security Groups, and User-Defined Routes
  - Azure Key Vault
  - Azure Automation
  - Public IP
  - Azure Private Link Service
- Have an Azure subscription and necessary permissions.

Migration Considerations

Data Migration Types

During migration, you need to consider the following data types:

- **Blob Storage**: For object storage.
- **Azure Files**: For file sharing.
- **Table Storage**: For NoSQL data storage.
- **Queue Storage**: For message queues.

Possible Dependencies

Ensure all services and applications dependent on the source storage account can correctly access the new storage account. This includes but is not limited to:

- Azure Web Apps
- Virtual Machines
- Azure Kubernetes Service
- Azure Container Instances
- Azure Functions
- Azure Batch
- Azure Synapse Analytics
- Azure Databricks
- Other Azure services using the storage account

Migration Timing

- **Planning Window**: Choose a time with the least business impact to perform the migration, preferably during off-peak hours or maintenance windows.
- **Estimated Time**: Estimate the time required for the entire migration based on the amount of data in the storage account. Large scale data migrations can take from hours to days.

Downtime Impact

- **Planned Downtime**: If downtime is allowed, you can pause or shut down all services dependent on the storage account during migration to minimize data inconsistency risks.
- **Minimized Downtime**: For scenarios requiring business continuity, design a migration strategy to ensure minimal downtime, such as using read-write permissions switching or setting up temporary forwarding mechanisms.

Non-Downtime Incremental Data Migration

- **Initial Sync**: First, complete a full backup and replication of data to the storage account in the target region.
- **Incremental Sync**: After the full migration, use tools like Azure Data Factory, AzCopy with `/Sync` parameter, or Event Grid with function-triggered replication logic to capture and migrate incremental data changes, ensuring no new data is lost during the migration.
- **Final Consistency**: Determine a final switch-over point, stop write operations to the source storage account, perform the last incremental sync, and then point all services to the new storage account.

Testing and Validation

- **Checksum Comparison**: Before and after migration, use the checksumming functions provided by storage services (e.g., Blob MD5 or File Content-MD5) to verify the integrity of data blocks, ensuring data consistency between source and target storage accounts.
- **Sample Validation**: Randomly select portions of data for detailed comparison, including metadata and Access Control Lists (ACLs), ensuring this information remains intact or unmodified during migration.
- **Full Data Scan**: For critical and manageable amounts of data, perform a thorough comparison to ensure no data is missed.

Cost Considerations

Evaluate potential additional costs during and after the migration, including data network transfer fees, storage fees, and potential compute resources fees (for data processing and migration).

Migration Solutions

Storage Account

- Azure Storage Account - Relocating an Azure Storage Account to Another Region

Blob Storage

For Blob containers

- Azure Blob - Use AzCopy to Copy Blobs Between Azure Storage Accounts
- Use Azure Data Factory to Copy Data to/from Azure Blob Storage

For Azure Unmanaged Disks

- As a best practice, Azure recommends converting unmanaged disks to managed disks. Refer to the instructions on Migrate your Azure unmanaged disks by September 30, 2025.
- Refer to Migrating Azure VMs to Managed Disks

Azure Data Lake Storage Gen2

- Refer to AzCopy for copying between storage accounts.
- Refer to Copy and transform data in Azure Data Lake Storage Gen2 using Azure Data Factory or Azure Synapse Analytics

Azure Files

- Migrate files from one SMB Azure file share to another
- Copy data from or to Azure Files by using Azure Data Factory

Table Storage

As a best practice, we recommend migrating your Table Storage to Azure Cosmos DB for Table accounts. However, if you need to continue using Table Storage, you can follow the subsequent steps for data migration.

- Migrate your data to an Azure Cosmos DB for Table account
- Copy data to and from Azure Table storage using Azure Data Factory or Synapse Analytics

Copying Using AzCopy

The latest known version supporting Table Storage copy is AzCopy V8. Later versions do not support copying Azure Table Storage. Refer to the following Azure Storage Explorer migration method.

Migration Using Azure Storage Explorer

Prerequisites

- Installed the latest version of Azure Storage Explorer

Step 1: Connect to Source and Target Storage Accounts

1. Open Azure Storage Explorer.
2. In the left-hand resource explorer, click “Add Account” or “Connect to Azure Storage”.
3. Choose the appropriate connection method (e.g., via Azure Active Directory, using storage account name and key, etc.).
4. Connect to the source storage account and the target storage account sequentially.

Step 2: Migrate Table Data

1. In Azure Storage Explorer, expand the source storage account and find the Table to be migrated.
2. Right-click the Table to be migrated and select “Copy Table”.
3. Expand the target storage account and find Tables under it.
4. Right-click Tables and select “Paste Table”.
5. Monitor the activity prompts until you see information like

```
Successfully copied table “source/tableName” to “target/tableName”; copied n entities
```

Queue Storage

When migrating Queue Storage, it is recommended to create a new queue in the new region and switch applications to write messages to the new queue. Once the old queue data is entirely consumed, switch the application’s queue read configuration to the new queue.

Related Reference Documents

- Choose an Azure solution for data transfer
- Optimize the performance of AzCopy with Azure Storage
- Manage Storage Account Local Users with Azure CLI
- Map a custom domain to an Azure Blob Storage endpoint

Migrate Azure Event Hubs

Table of Contents

- Azure Event Hubs Regional Migration Guide
  - Table of Contents
  - Introduction
  - Prerequisites
  - Service Endpoint Considerations
  - Private Endpoint Considerations
  - Migrate Namespace
  - Migrate Dedicated Cluster

Introduction

This guide is designed to instruct you on how to migrate Azure Event Hubs from one region to another.

You cannot directly migrate Azure Event Hubs resources across Azure regions. The Event Hubs service does not have data export or import functionality. You can export Event Hubs resources to Resource Manager templates, adjust the exported templates for the target Azure region, and then recreate the resources.

Prerequisites

- Ensure that the services and features used by your account are supported in the target region.
- If you have enabled capture functionality for Event Hubs in the namespace, first move the Azure Storage or Azure Data Lake Store Gen 2 account before moving the Event Hubs namespace. You can also move the resource group containing the storage and Event Hubs namespace to another region following similar steps as those described in this article.
- If the Event Hubs namespace is in an Event Hubs cluster, move the dedicated cluster to the target region before following the steps in this article. You can also use the QuickStarts template on GitHub to create an Event Hubs cluster. In the template, remove the namespace part in the JSON to create only the cluster.
- Identify all resource dependencies. Depending on how Event Hubs is deployed, the following services might need to be deployed in the target region:
  - Public IP
  - Virtual Network
  - Event Hubs namespace
  - Event Hubs cluster
  - Storage account

Tip
Once the capture feature is enabled, you can migrate the storage account from the source region or use an existing storage account in the target region.

- Identify all dependent resources. Event Hubs is a messaging system that allows applications to publish and subscribe to messages. Consider whether the application in the target region requires the same set of dependent services for messaging support as in the source region.

Service Endpoint Considerations

Azure Event Hubs' virtual network service endpoints can restrict access to specified virtual networks. Additionally, these endpoints can restrict access to a range of IPv4 (Internet Protocol version 4) address ranges. Users connecting to Event Hubs from outside cannot access these resources. If service endpoints are configured in the source region of the Event Hubs resources, the same needs to be configured in the target region.

To successfully recreate Event Hubs in the target region, you must pre-create the VNet and subnets. If you use the Azure Resource Mover tool to move all these resources, service endpoints will not be automatically configured. Thus, you need to manually configure service endpoints, which can be done through the Azure Portal, Azure CLI, or Azure PowerShell.

Private Endpoint Considerations

Azure Private Link provides private connectivity from a virtual network to Azure platform as a service (PaaS), customer-owned services, or Microsoft partner services. Private Link simplifies the network architecture and secures the connection between endpoints in Azure by eliminating exposure of data over the public Internet.

To successfully recreate Event Hubs in the target region, you must pre-create the VNet and subnets before the actual recreation.

Migrate Namespace

To migrate the namespace to another region, refer to the official documentation: Move an Azure Event Hubs namespace to another region

Migrate Dedicated Cluster

To migrate a dedicated cluster to another region, refer to the official documentation: Move an Azure Event Hubs dedicated cluster to another region

Migrate Azure Notification Hubs

Table of Contents

- Introduction
- Preparation
- Migration Steps
  - Stop Notification Hubs Service
  - Export Notification Hubs Data
  - Create Notification Hubs in Target Region
  - Import Notification Hubs Data to Target Region
  - Update Service Connection Strings
- Validation and Testing
- Cleanup Resources
- Conclusion

Introduction

This guide is designed to help you migrate Azure Notification Hubs from one region to another. The entire process ensures data integrity to avoid service interruptions and data loss.

Preparation

Before starting the migration, make sure you have the following prerequisites: 1. A valid Azure subscription. 2. Administrative access to both source and target regions. 3. Azure CLI installed and configured. 4. Permissions to create necessary resource groups in both source and target regions.

Migration Steps

Stop Notification Hubs Service

First, to ensure data consistency, it is recommended to pause services that implicitly use Notification Hubs before migration.

Export Notification Hubs Data

Export the namespace and Hub configuration data of the Notification Hubs to a local file.

```
# Export namespace configuration
az notification-hub namespace show --resource-group <ResourceGroupName> --namespace-name <NamespaceName> > namespace_config.json

# Export Hub configuration
az notification-hub show --resource-group <ResourceGroupName> --namespace-name <NamespaceName> --name <HubName> > hub_config.json
```

Create Notification Hubs in Target Region

Create a new Notification Hubs namespace and Hub in the target region.

```
# Create namespace
az notification-hub namespace create --resource-group <TargetResourceGroupName> --name <TargetNamespaceName> --location <TargetRegion>

# Create Hub
az notification-hub create --resource-group <TargetResourceGroupName> --namespace-name <TargetNamespaceName> --name <TargetHubName> --properties hub_config.json
```

Import Notification Hubs Data to Target Region

Import the configuration data exported from the source region into the Notification Hubs of the target region.

```
# Transfer repository connection strings and other required configuration information to the target Hub
az notification-hub update --resource-group <TargetResourceGroupName> --namespace-name <TargetNamespaceName> --name <TargetHubName> --properties hub_config.json
```

Update Service Connection Strings

All applications and services using the Notification Hubs service will need to update their connection strings to point to the new region.

Validation and Testing

1. **Test the service**: Send a push notification to ensure that the newly created Notification Hubs service is working properly. bash az notification-hub test-send --resource-group <TargetResourceGroupName> --namespace-name <TargetNamespaceName> --name <TargetHubName> --payload "{'message':'test notification'}"
2. **Application validation**: Confirm that all affected services and applications have successfully updated the connection strings and that the notification service is functioning normally.

Cleanup Resources

After confirming a successful migration and that the new Notification Hubs service is functioning correctly, old Notification Hubs resources in the source region can be deleted to save costs.

```
az notification-hub delete --resource-group <ResourceGroupName> --namespace-name <NamespaceName> --name <HubName>
az notification-hub namespace delete --resource-group <ResourceGroupName> --name <NamespaceName>
```

Conclusion

By following the above steps, you have successfully migrated Azure Notification Hubs from one region to another. During the migration process, carefully review each step to ensure data integrity and service continuity. If you encounter any issues during the migration, refer to the official Azure documentation or contact Azure technical support.

Learn more about Azure Notification Hubs by referring to the following resources: * Azure Notification Hubs Documentation

For any questions, please contact your Azure support team.

Migrate Azure Service Bus

In many cases, it may be necessary to move an existing Service Bus namespace from one region to another. For example, you might need to create a namespace with the same configuration for testing purposes. Additionally, you may need to create another namespace in a different region as part of a disaster recovery plan.

Here are the summary steps:

1. Export the Service Bus namespace from the current region to an Azure Resource Manager template.
2. Update the location of the resources in the template. Also, remove the default subscription filters from the template, as users cannot create default rules; the system will create them automatically.
3. Use the template to deploy the Service Bus namespace to the target region.
4. Verify the deployment to ensure the namespace, queues, topics, and topic subscriptions have been created in the target region.
5. Update the AccessKey used by the service bus service of the source region to the AccessKey of the target region.
6. Delete the namespace from the source region and complete the move.

Prerequisites

Ensure that the target region supports the Azure Service Bus and features used by the account.

Preparation

To start, export the Resource Manager template. This template contains the settings that describe the Service Bus namespace.

1. Sign in to the Azure portal.
2. Select “All resources” and then choose your Service Bus namespace.
3. On the “Service Bus Namespace” page, select “Export template” under “Automation” from the left-side menu.
4. In the “Export template” page, select “Download”.
5. Locate the .zip file downloaded from the portal and extract it to a folder of your choice. This zip file contains the template and parameter JSON files.
6. In the extracted folder, open the template.json file.
7. Search for `location` and replace the value of this property with the new name of the region or location. The region code is the name of the region without spaces, such as `chinaeast` for `China East`.
8. Remove the following types of resource definitions: `Microsoft.ServiceBus/namespaces/topics/subscriptions/rules`. Don’t forget to remove the preceding comma (,) character before this section to ensure valid JSON.

Note: You cannot use Resource Manager templates to create default rules for subscriptions. When creating subscriptions in the target region, the system will automatically create default rules.

Move

Deploy the template to create the Service Bus namespace in the target region.

1. Use “Template Deployment (Deploy using custom template)” to redeploy the service bus. The template uses the template.json file downloaded in the previous step.

2. On the Customize Deployment page, select the target region to be deployed, enter the name of the new service bus namespace, and click Create.

Verify

After the deployment is successful, select “Go to resource group”. On the “Resource group” page, select “Service Bus Namespace”.

On the “Service Bus Namespace” page, verify that you can see the queues, topics, and subscriptions from the source region.

1. In the namespace, you can see “Queues” at the bottom of the right pane.

2. Switch to the “Topics” tab to view the topics in the namespace.

3. Select a topic to verify that subscriptions have been created.

Migrate Azure IoT Hub

This article describes how to migrate an IoT hub to a new region.

To migrate a hub, you need a subscription with management access to the original hub. You can place the new hub in a new resource group and region, the same subscription as the original hub, or even a new subscription. You cannot use the same name because hub names must be globally unique.

Considerations

There are several things to consider before migrating an IoT hub.

- Ensure that all available features in the original location are also available in the new location. Some services are offered in preview, so not all features are available in every location.
- Do not delete the original resources until you have created and verified the version you are migrating. Once a hub is deleted, it is permanently gone and there is no way to restore it, meaning you can't verify settings or data to ensure the hub is properly replicated.
- Data in the original Azure IoT hub will not be migrated. This data includes device messages, cloud-to-device (C2D) commands, and job-related information such as schedules and history. Metrics and logging results will also not be migrated.
- Prepare for the downtime caused by the migration. Cloning devices to the new hub will take some time. If using the import/export method, benchmarking shows moving 500,000 devices can take about two hours, and moving 1 million devices can take around four hours.
- Devices can be copied to the new hub without shutting them down or making changes.
  - If the devices were initially provisioned using DPS, update their registration to point to the new IoT hub. Then, re-provision the devices to update the connection info stored in each device.
  - Otherwise, you must use the import/export method to move the devices, after which you'll need to modify the devices to use the new hub. For example, you can set the devices to use the IoT hub hostname in the twin desired properties. Devices will adopt this IoT hub hostname, disconnect from the old hub, and reconnect to the new hub.
- Update any certificates so that they can be used for the new resources. Additionally, you may have defined the hub somewhere in the DNS table, and this DNS information will need to be updated.

Method

Here is the recommended general method for migrating an IoT hub.

1. Export the hub and its settings to a resource manager template.
2. Make the necessary changes to the template, such as updating all occurrences of the name and location to reflect the migrated hub. For any resource used in message routing endpoints in the template, update the resource keys in the template.
3. Import the template into a new resource group located in the new region. This step will create the new IoT hub.
4. Debug as needed.
5. Add anything not exported to the template.

For example, consumer groups are not exported to the template. You need to manually add consumer groups to the template or add them using the Azure portal after creating the hub.

6. Copy devices from the original hub to the new hub. This process is described in the Manage the devices registered to the IoT hub section.

How to Handle Message Routing

If your hub uses message routing, the process of exporting the hub template involves routing configuration, but not the resources themselves. If you are migrating the IoT hub to a new region, you must decide whether to move the routing resources to the new location or leave them in place and continue to use them “as is”. Routing messages to endpoint resources in a different region may lead to slight performance degradation.

If the hub uses message routing, you can take one of two approaches.

- Move the resources used for routing endpoints to the new location.

    1. Create new resources in the Azure portal or by the resource manager template manually.

    2. All resources created in the new location need to be renamed since they need globally unique names.

    3. Update the resource names and resource keys in the template for the new hub before creating it. These resources should exist when the new hub is created.

- Do not move the resources used for routing endpoints. Use these resources “in place”.

    1. During the template editing step, retrieve the key for each routing resource and place it into the template before creating the new hub.

    2. The hub will still reference the original routing resources and route messages to these resources as configured. Performance will degrade slightly because the hub and routing endpoint resources are not in the same location.

Preparing to Migrate the Hub to Another Region

This section provides specific instructions on migrating the hub.

1. Export the original hub to a resource template and save the file to a location where it can be easily found again.

2. Modify the template content. Locate the file named `template.json` in the downloaded template files, and make the following modifications according to your specific needs:

- If there are no associated containers, delete the container name parameter section at the top.

- Delete the `storageEndpoints` property.

- Change the `location` property under `resources` to the target region.

Update Routing Endpoint Resources

When you export the resource manager template for a hub with routing configured, you’ll find that the keys for these resources are not included in the exported template. They are indicated by asterisks. You must go into the portal to retrieve the keys for these resources and populate them before importing and creating the new hub.

If you are also moving the routing resources, update the name, ID, and resource group for each endpoint as well.

1. Retrieve the necessary keys for all routing resources and place them into the template. You can retrieve the keys from the resources in the Azure portal.

   – For example, if routing messages to a storage container, find the corresponding storage account in the portal. Under the “Settings” section, select “Access Keys” and copy one of the keys. When the template is initially exported, the key appears as follows:

   ```
   "connectionString": "DefaultEndpointsProtocol=https;AccountName=fabrikamstorage1234;Accou
   ntKey=****",
   "containerName": "fabrikamresults",
   ```

   After retrieving the storage account’s account key, input it into the clause in the template, replacing the asterisks.

   – For Service Bus queues, get the shared access key that matches `SharedAccessKeyName`. Here is the key and `SharedAccessKeyName` in JSON:

   ```
   "connectionString": "Endpoint=sb://fabrikamsbnamespace1234.servicebus.chinacloudapi.cn:5671/;
   SharedAccessKeyName=iothubroutes_FabrikamResources;SharedAccessKey=****;EntityPath=fabri
   kamsbqueue1234",
   ```

   – The same steps apply for Service Bus topics and Azure Event Hub connections.

Create the New Hub by Loading the Template

Use the edited template to create the new hub. If you are moving routing resources, set up the resources in the new location and update the references in the template to match. If you are not moving routing resources, include them in the template and use the updated keys.

1. Sign in to the Azure portal.
2. Select “Create a resource”.
3. In the search box, search and select “Template deployment (deploy using custom templates)”. On the template deployment screen, select “Create”.
4. On the “Custom deployment” page, choose “Build your own template in the editor” to upload the template from a file.
5. Select “Load file”.
6. Browse to and select the edited template, then select “Open”. The template will be loaded into the editor window. Select “Save”.
7. On the “Custom deployment” page, fill in the following fields.

   Subscription: Choose the subscription to use.

   Resource Group: Select an existing resource group or create a new resource group.

   Region: If you selected an existing resource group, the region field will auto-populate to match the location of the resource group. If creating a new resource group, this is the location for that resource group.

   Connection String: Populate the connection string for the hub.

   Hub Name: Name the new hub.

8. Select the “Review + create” button.

9. Select the “Create” button. The portal will validate your template and deploy the new hub. If there are routing configuration data, it will be included in the new hub but will point to resources in the previous location.

Move Devices to the New Hub Using the Import/Export Method

You can use the IoT C# sample: Azure IoT SDK for C# to import and export devices.

For detailed steps, refer to: How to manually migrate IoT Hub - Azure IoT Hub

Viewing Results

You can view the devices in the Azure portal and verify that they appear in the new location.

1. Go to the new hub using the Azure portal. Select the hub and then select “IoT devices.” You will see the devices copied from the old hub to the new hub. You can also view the properties of the new hub.

2. Check import/export errors: In the Azure portal, go to the Azure storage account and then check the ImportErrors.log in the devicefiles container. If this file is empty (size 0), it means no errors occurred. If you try to import the same device multiple times, the second import will reject the device and add an error message to the log file.

Submitting Changes

At this point, you have copied the hub to a new location and migrated the devices to the new hub. Next, you need to make the necessary changes for the devices to work with the new hub.

To submit changes, follow these steps:

- Update each device to change the IoT Hub hostname to point to the new hub. This should be done using the same method that was used when the devices were first provisioned.

- Change all applications that reference the old hub to point to the new hub.

- Once completed, the new hub should be up and running. The old hub should have no active devices and be in a disconnected state.

Rolling Back Changes

If you decide to roll back the changes, follow these steps:

- Update each device to change the IoT Hub hostname to point to the old hub. This should be done using the same method that was used when the devices were first provisioned.

- Change all applications that reference the new hub to point to the old hub. For example, if you are using Azure Analytics, you may need to reconfigure the Azure Stream Analytics input.

- Delete the new hub.

- If you have routing resources, the configuration on the old hub should still point to the correct routing configuration, and it should still be able to use these resources after the old hub restarts.

Checking Results

To check the results, switch your IoT solution to point to the hub located in the new location, and then run it. In other words, perform the same operations on the new hub that were previously performed on the old hub and ensure they work correctly.

If you have implemented routing, test and ensure that messages are correctly routed to the resources.

Cleanup

Do not clean up resources until you have confirmed that the new hub is up and running and the devices are working properly. Additionally, if you have used routing functionality, be sure to test this functionality. Once ready, follow these steps to clean up the old resources:

- Delete the old hub (if you have not already done so). This will remove all active devices from that hub.
- If you have moved routing resources to the new location, you can delete the old routing resources.

Next Steps

You have now migrated the IoT Hub along with the devices to a new hub in a new region. For more information on performing bulk operations on the identity registry within the IoT Hub, see Bulk import and export IoT Hub device identities.

Migrate Azure Stream Analytics Jobs

The simplest way to migrate Azure Stream Analytics jobs across Azure regions is by using the Visual Studio Code for Azure Stream Analytics to copy the job to another region. > **Note:**
> The ASA (Azure Stream Analytics) tool extension for Visual Studio is no longer maintained. It is recommended to use the ASA tool extension in Visual Studio Code for this task.

You can also manually redeploy the service in the target Azure region using the Azure portal or PowerShell. The input and output sources for a stream analytics job can be located in any region.

You can follow the steps below to migrate Azure Stream Analytics using the Visual Studio Code tool.

Prerequisites

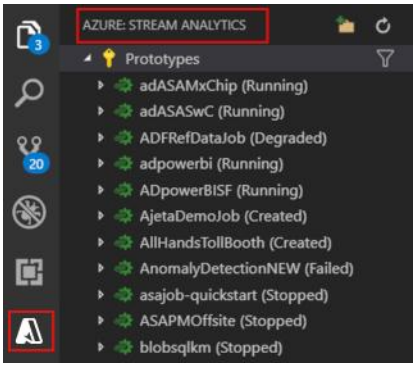
Before you begin, ensure that you have the following:

1. A valid Azure subscription.
2. Permissions for both the source and target regions.
3. Install Visual Studio Code.
4. Install the Azure Stream Analytics extension for Visual Studio Code.
5. Sign in to Azure in China using Visual Studio Code.

Migration Steps

Displaying the Stream Analytics Job in Visual Studio Code

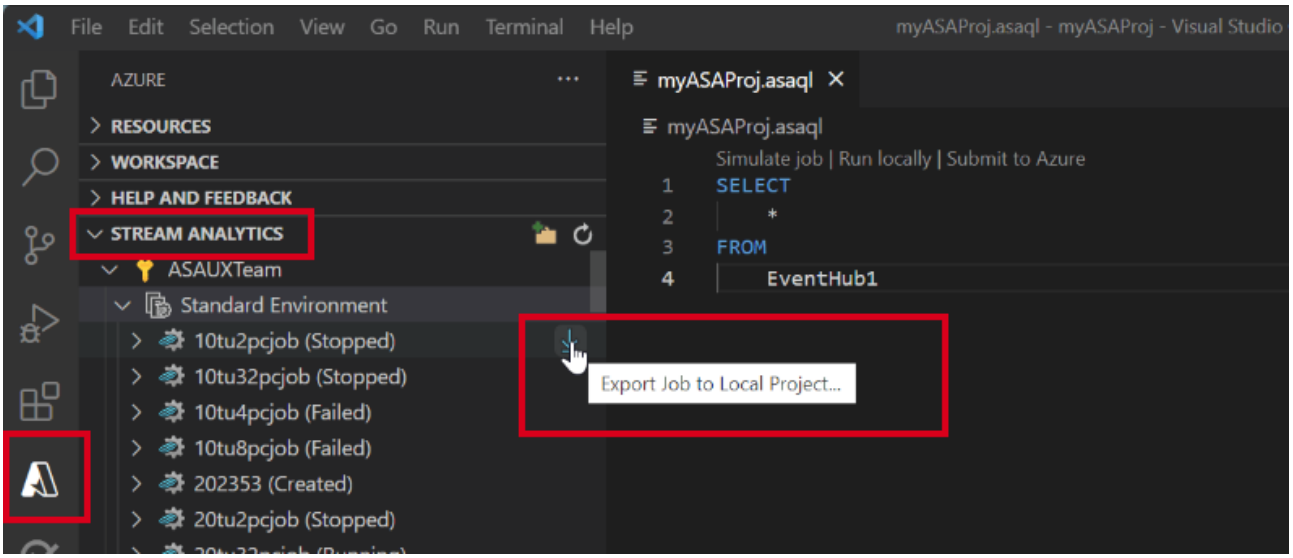
Select the “Azure” icon from the Visual Studio Code activity bar and expand the “Stream Analytics” node. Find the job that needs to be migrated.



Open Stream Analytics Explorer

Export Configuration and State

To export a job to a local project, find the job you want to export in the Stream Analytics Explorer in Visual Studio Code. Then select a folder for the exported project.



Find the ASA Job in Visual Studio Code

The project will be exported to the selected folder and added to the current workspace.

Create a Resource Group in the Target Region

Create a resource group in the target region.

Duplicate Input and Output Resources

Ensure that the input and output resources in the source and target regions are consistent. The specific steps depend on the types of input and output you are using, such as Blob storage, Event Hub, SQL Database, etc. Create these resources as needed.

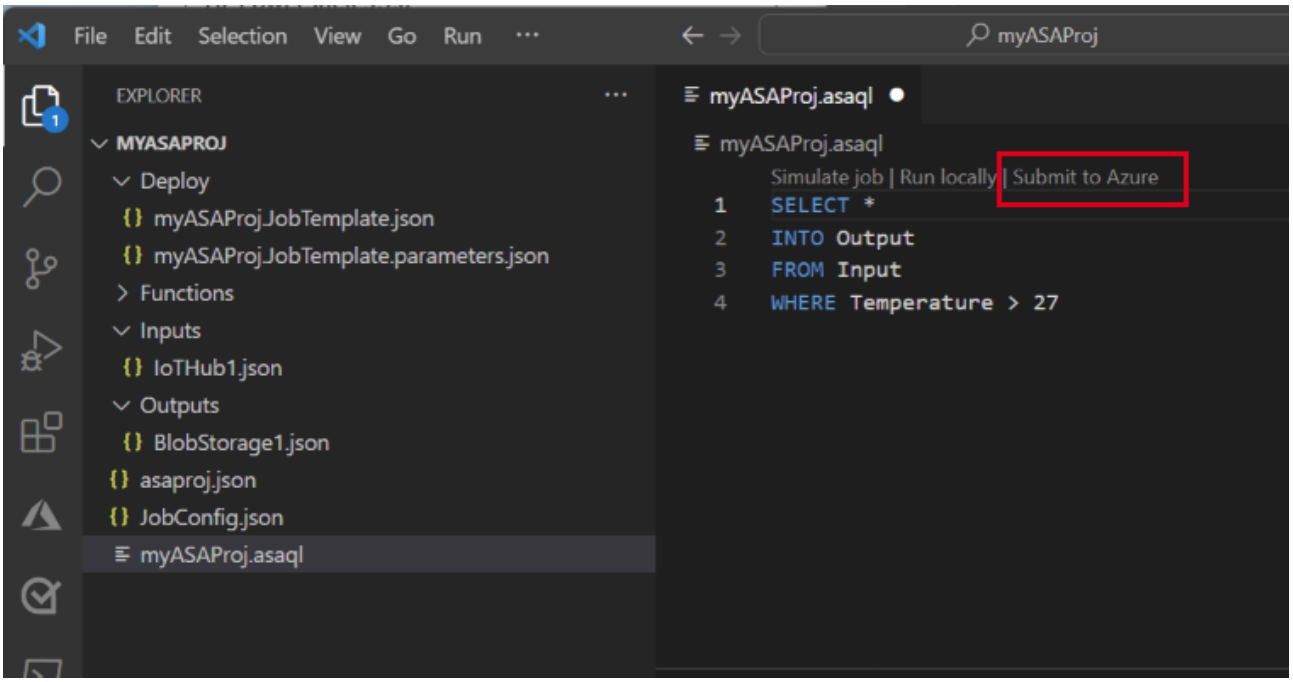
Create a New Azure Stream Analytics Job

Create a new Azure Stream Analytics job in the target region. If you wish to create the job in Azure Stream Analytics at the target during the import process, you can skip this step and proceed to the next step.

Import Configuration to the Target

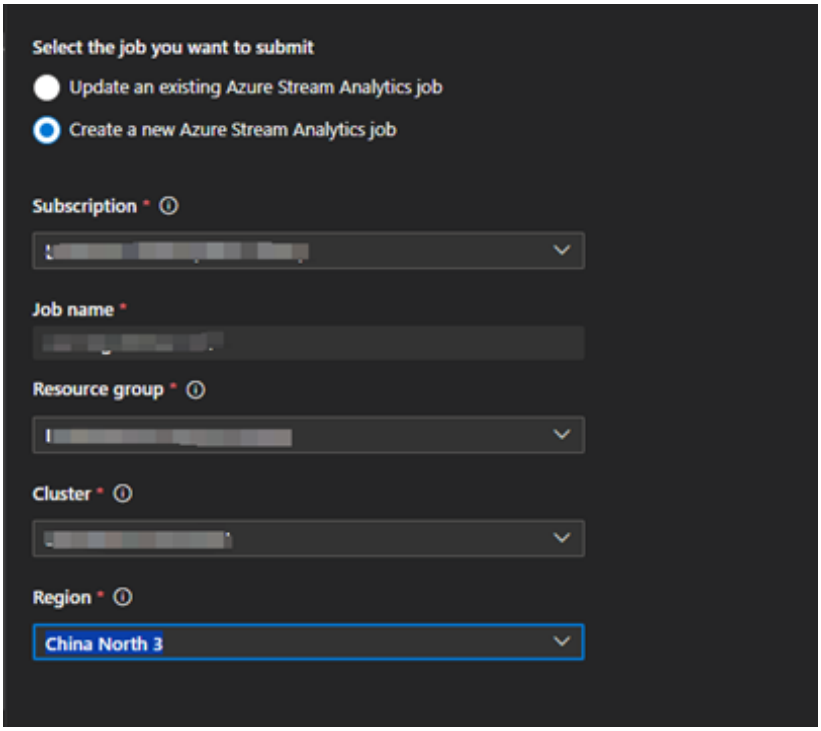
In the query editor in Visual Studio Code, find the project files exported in the earlier steps, click the (*.asaql) file, and select “Submit to Azure,” then follow the instructions.

In this operation, you can either import the configuration into an existing Azure Stream Analytics job at the target or choose to create a new Azure Stream Analytics job and import the configuration.



Import Configuration

If you choose to create a new Azure Stream Analytics job and import the configuration, make sure to select your target region.



Select Job

Stop the Original Azure Stream Analytics Job

After completing the related work and checking and updating the configurations in the target region, you can stop the Azure Stream Analytics job in the original region via the Azure portal and prepare to start the Azure Stream Analytics job in the target region.

Start the New Azure Stream Analytics Job

Start the new Azure Stream Analytics job to begin data processing.

Verify and Test

1. Check the status and running condition of the new job through the Azure portal or Azure CLI.
2. Confirm that the input and output data streams are being processed correctly.
3. Monitor the job for some time to ensure there are no errors or data loss in the data processing.

Clean Up Resources

If the new Azure Stream Analytics job is confirmed to be running normally, you can delete the old job and related resources in the source region.

Summary

By following the above steps, you have successfully migrated an Azure Stream Analytics job from one region to another. Ensure to carefully check each step during the migration to avoid any data loss or service interruption. If you encounter any issues, refer to the Azure official documentation or contact Azure technical support.

For more information: * Refresh your knowledge by completing Stream Analytics Tutorials. * Refer to the Stream Analytics Overview. * Learn how to create a Stream Analytics job using PowerShell.

HDInsight

HDInsight does not support migration from one region to another. We recommend creating and configuring HDInsight in the new region.

Prerequisites

- Ensure that your Azure subscription allows the creation of HDInsight and its corresponding resources in the target region.
- Understand all the services required by HDInsight before planning the migration strategy. Appropriate migration strategies must be selected for the services involved in the migration.

Metadata Migration

Backup and Restore

For HDInsight metadata stored in an Azure SQL database, use Azure SQL Database's backup and restore functionality. Refer to Restore a database using automatic database backups in Azure SQL Database.

In the target source database, modify the source data storage address to the new data storage address in the corresponding fields and tables.

Metadata Script Migration

Use scripts to migrate Hive metastore:

1. Generate Hive DDL from the local Hive metastore. You might consider using a wrapper bash script for this step.
2. Edit the generated DDL to modify the source data storage address to the new data storage address.
3. Run the updated DDL against the metastore in the HDInsight cluster.
4. Ensure that the Hive metastore versions between the source and target are compatible.

Migration

Once everything is verified to be working as expected, schedule downtime for the migration. During this downtime, perform the following actions:

1. Back up all temporary data stored on local cluster nodes.
2. Stop the HDInsight cluster.
3. Use AzCopy or a similar tool to migrate data from the Azure storage account to the new region.
4. Create new HDInsight resources in the target Azure region. Refer to Learn how to create Linux-based HDInsight clusters. Attach the migrated storage resources as the primary connection storage.
5. Import any backed-up temporary data.
6. Rebuild objects using DDL statements.
7. Start jobs/continue processing using the new cluster.

Workload-Specific Guidelines

The following documents provide guidelines on how to migrate specific workloads:

- Migrate HBase
- Migrate Kafka

- Migrate Hive/Interactive Query

Verification

Perform verification tests to ensure that jobs are working as expected on the new cluster.

For More Information

- Refer to the Azure HDInsight documentation.
- Refresh your knowledge by completing the HDInsight tutorials.
- For help with scaling HDInsight clusters, see Administer HDInsight using PowerShell.
- Learn how to use AzCopy.

Migrate Azure Logic Apps

Overview

Azure Logic Apps is a cloud platform where you can create and run automated workflows with minimal coding. By using the visual designer and selecting from pre-built operations, you can quickly build workflows to integrate and manage apps, data, services, and systems. Azure Logic Apps allows you to create either “Consumption” or “Standard” logic app workflows.

This document aims to guide you on how to migrate Azure Logic Apps from one region to another.

Migration Preparation

Before starting the migration, ensure you have the following:

- The same Azure subscription that was used to create the logic apps or integration accounts you wish to move.
- Resource owner permissions required to move and set up resources. Learn more about Azure Role-Based Access Control (Azure RBAC).

Considerations

- You can only move specific types of logic app resources between Azure resource groups or subscriptions.
- Check the limits on the number of logic app resources that can be used in each Azure subscription and region here. These limits affect the ability to move specific resource types when the regions of different subscriptions or resource groups are the same. For example, only one Free tier integration account can be used in each Azure region per Azure subscription.
- When moving resources, Azure will create new resource IDs. Therefore, ensure you use the new IDs and update any scripts or tools associated with the resources that you are moving.
- After migrating logic apps between subscriptions, resource groups, or regions, you must recreate or reauthorize any connections that require Open Authentication (OAuth).
- Integration Service Environments (ISE) can only be moved to another resource group within the same Azure region or Azure subscription. ISE cannot be moved to a resource group in another Azure region or Azure subscription. Additionally, after such moves, you must update all references to the ISE in your logic app workflows, integration accounts, connections, etc.

Migration Scenarios

- For more information, read the Move logic app resources to other Azure resource groups, regions, or subscriptions document, specifically the **Move resources across regions** section.

Related References

- Get familiar with Azure Logic Apps by completing the Logic App tutorials.
- Read the Azure Logic Apps Overview.

Azure Scheduler Jobs

Azure Scheduler has been fully retired as of January 31, 2022.

For more information, read:

- Azure Scheduler will be retired on 31 January, 2022
- Migrate Azure Scheduler jobs to Azure Logic Apps

Migrate Media Services

Table of Contents

- Introduction
- Preparations
- Execute Migration
- Cleanup Resources
- Summary

Introduction

Azure Media Services was discontinued on June 30, 2024. The Media Services retirement guide provides options for migrating to solutions from Microsoft's partner ecosystem or other Azure services. For details, please see Azure Media Services Retirement Guide.

Preparations

Before starting the migration, ensure you have the following:

1. A valid Azure subscription.
2. Permissions for the migration account in the source region.
3. No ongoing tasks for the source media services, and essential data has been backed up.
4. Comprehensive analysis of Microsoft partner media services and pre-validation of migration plans.

Execute Migration

Perform the migration according to the selected Microsoft partner media services and the formulated migration plan.

Cleanup Resources

If the migrated media services are confirmed to be running normally, you can delete the old media service account and related resources in the Azure source region.

Summary

Please migrate to solutions offered by Microsoft partners suitable for on-demand encoding, live video streaming, on-demand streaming, and content protection by June 30, 2024. If you encounter any problems, refer to the official Azure documentation or contact Azure technical support.

Refer to the following documents for more detailed information:

- Azure Media Services Tutorial
- Azure Media Services Retirement Guide

If you have any questions, please contact your Azure support team.

Migrate Azure Site Recovery

Overview

You can’t move an existing Azure Site Recovery setup across Azure regions.
Azure Site Recovery itself is used for recovering VMs from the source region, Region1, to the target region, Region2. You need to create the Azure Recovery Vault in the target region, Region2.
When you decide to migrate the Azure Site Recovery region, it essentially means changing the target region for VM recovery, so the configurations that existed in the previous target region cannot be reused.

You can disable the existing configuration and set up a new site recovery solution in the target Azure region.
For more information, please see Relocate Azure Recovery Vault and Site Recovery to another region.

Summary

Before migrating in a production environment, test and validate in a test environment.
If you encounter any issues, refer to Azure official documentation or contact Azure technical support.

For more information:

* Azure to Azure disaster recovery
* VMware to Azure disaster recovery
* Hyper-V to Azure disaster recovery
* Physical to Azure disaster recovery