Install and configure Red Hat Ansible Automation Platform on Microsoft Azure
Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

Providing Feedback: If you have a suggestion to improve this documentation, or find an error, create an issue at https://issues.redhat.com. Select the Ansible Automation Platform project and use the Docs component.
Table of Contents

PREFACE ................................................................................................................................. 3

CHAPTER 1. INTRODUCTION TO ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE ....... 4
  1.1. ABOUT RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE ............ 4
  1.2. APPLICATION ARCHITECTURE ................................................................................. 4
      1.2.1. Ansible Automation Platform on Microsoft Azure infrastructure usage ............... 5
      1.2.2. Lifecycle management ....................................................................................... 6
      1.2.3. Ansible Automation Platform on Microsoft Azure scaling .................................. 6
      1.2.4. Migration ........................................................................................................... 6

CHAPTER 2. INSTALLING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE ...... 7
  2.1. PREREQUISITES ................................................................................................. 7
      2.1.1. Verifying your Red Hat registry credentials ....................................................... 7
      2.1.2. Azure resource quotas and infrastructure limits ............................................... 7
      2.1.2.1. Regional vCPU limits ..................................................................................... 7
      2.1.2.2. Regional StandardCore limits ...................................................................... 8
  2.2. CREATING A SERVICE PRINCIPAL ..................................................................... 8
      2.2.1. Maintaining your service principals ................................................................. 9
  2.3. DEPLOYING ANSIBLE AUTOMATION PLATFORM FROM AZURE MARKETPLACE .......... 9
      2.3.1. Locating Ansible Automation Platform in Azure Marketplace ....................... 9
      2.3.2. Provisioning Red Hat Ansible Automation Platform on Microsoft Azure .......... 10
  2.4. ACCESSING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE ...... 11
      2.4.1. License association ......................................................................................... 11
      2.4.2. Azure Active Directory (Azure AD) SSO configuration .................................... 11

CHAPTER 3. PRIVATE NETWORK PEERING ......................................................................... 15
  3.1. HUB-AND-SPOKE PEERING (TRANSIT ROUTES) ...................................................... 15
      3.1.1. Hub-and-spoke peering process overview ....................................................... 15
      3.1.1.1. Finding the CIDR Block of the managed application Kubernetes cluster .......... 16
      3.1.1.2. Configuring network peering with the Ansible Automation Platform subnet .... 16
      3.1.1.3. Updating the route tables ............................................................................. 17
  3.2. DIRECT PEERING .............................................................................................. 18
      3.2.1. Configuring direct network peering ................................................................. 18
Thank you for your interest in Red Hat Ansible Automation Platform. Ansible Automation Platform is a commercial offering that helps teams manage complex multi-tier deployments by adding control, knowledge, and delegation to Ansible-powered environments.

This guide helps you to understand the installation and use of Ansible Automation Platform on Microsoft Azure apps on cloud platforms. This document has been updated to include information for the latest release of Ansible Automation Platform on Microsoft Azure.
CHAPTER 1. INTRODUCTION TO ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE

1.1. ABOUT RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE

Red Hat Ansible Automation Platform on Microsoft Azure is a managed application that you can deploy from the Azure Marketplace portal to a resource group in your Azure tenant. Ansible Automation Platform on Microsoft Azure provides access to a library of Ansible content collections, and it is integrated with key Azure services, so you can start deploying, configuring, and managing infrastructure and applications quickly.

The following Red Hat Automation Platform components are available on Ansible Automation Platform for Azure:

- Automation Controller
- Automation Hub
- Private Automation Hub
- Automation Service Catalog
- Ansible Content Collections, including the Microsoft collection for Azure
- Automation Execution Environment
- Ansible content tools, including access to Red Hat Insights for the Ansible Automation Platform

NOTE

Automation mesh is not available on Ansible Automation Platform for Azure.

1.2. APPLICATION ARCHITECTURE

Red Hat Ansible Automation Platform on Microsoft Azure is installed as a managed application. Red Hat manages both the underlying Azure resources and the software running on it while that infrastructure runs in your Azure tenant.

The managed application resource group is completely separate from other resource groups in your tenant. Red Hat only has access to the managed application resource group, with no visibility into other tenant resources.

For information about how this works and how resources and access are isolated from the rest of your Azure resources, refer to Azure managed applications overview in the Microsoft Azure managed applications guide.

Ansible Automation Platform on Microsoft Azure uses two resource groups:

- A resource group in your tenant. This resource group includes a single resource referring to the Ansible Automation Platform on Microsoft Azure managed app deployment. Red Hat has access to this managed app to perform support, maintenance, and upgrades.

- A multi-tenant managed resource group that contains all the infrastructure needed to operate
Ansible Automation Platform on Microsoft Azure. This multi-tenant resource group is shared between the Red Hat tenant and your tenant. Red Hat has full administrative control and you have read-only access to the resource group.

The following diagram outlines the application resources and architecture that are deployed into the managed application resource group when Ansible Automation Platform on Microsoft Azure is deployed into your Azure subscription.

1.2.1. Ansible Automation Platform on Microsoft Azure infrastructure usage

When you install Ansible Automation Platform on Microsoft Azure, the following infrastructure is deployed into your Azure subscription:

- Kubernetes Cluster
  - VM Shape: Standard_D4s_v3
  - Autoscaling Min Nodes: 1
  - Autoscaling Max Nodes: 20
- Azure Database Service (Postgres Database)
  - 128 GB Storage Allocation
- Storage
  - 1 Storage Account: StorageV2 - Standard_LRS
- Networking
Exact infrastructure usage depends on the length of time that the managed application is deployed in your tenancy and the automation requirements that may cause the Kubernetes cluster to autoscale to meet the demands of your workload.

1.2.2. Lifecycle management

Red Hat Ansible is responsible for the monitoring, health, and maintenance of the underlying services and {Ansible Automation Platform} core systems as well as the operation of Ansible Automation Platform on Microsoft Azure itself. This includes lifecycle management of the components.

1.2.3. Ansible Automation Platform on Microsoft Azure scaling

Ansible Automation Platform on Microsoft Azure default configuration of Microsoft Azure cluster autoscaler for autoscaling, with the following settings to limit the number of nodes: Minimum Nodes: 1 Maximum Nodes: 20

1.2.4. Migration

Red Hat does not provide a solution to migrate existing deployments to Ansible Automation Platform on Microsoft Azure.
CHAPTER 2. INSTALLING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE

2.1. PREREQUISITES

Azure requirements

- A subscription for Microsoft Azure.
- Contributor or Administrator access to that Azure subscription.
- Access to the Azure CLI.

Ansible Automation Platform requirements

- An account on the Red Hat Red Hat Customer Portal (access.redhat.com).
- A specific subscription entitlement for Red Hat Ansible Automation Platform.

2.1.1. Verifying your Red Hat registry credentials

You must provide Red Hat registry (registry.redhat.com) account information when provisioning Ansible Automation Platform for Azure. Verify your account credentials beforehand to avoid deployment failure.

1. Run the following command in a shell on a computer with Podman or Docker installed:

   ```bash
   podman login registry.redhat.io
   ```

2. Enter your Red Hat account username and password, and confirm that you can log in.

2.1.2. Azure resource quotas and infrastructure limits

Microsoft imposes resource limits within each Azure region. The CPU limit is the most likely to impact Red Hat Ansible Automation Platform on Microsoft Azure.

Before you install Ansible Automation Platform for Azure, ensure that you have capacity to deploy the managed application into your desired region. Refer to Azure infrastructure usage for infrastructure requirements.

2.1.2.1. Regional vCPU limits

The Azure resources used during the deployment of the managed application temporarily exceed the resource requirements in Azure infrastructure usage. The Total Regional vCPUs quota is temporarily consumed when deploying the managed application.

Every Azure region has a separate Total Regional vCPUs quota. To prevent installation failure, ensure that you have at least 8 available vCPUs in the Azure region where you want to deploy the managed application.

The following steps describe how to view the resource quotas for your subscription the Azure console:

1. In the Azure console, search for Quotas and open the My Quotas page.
2. Select the region where you wish to deploy the managed application to view your allocation and usage metrics for that region. Ensure that you have selected a single region. Viewing all regions at once will not show the limitations of a single Azure region.

2.1.2.2. Regional StandardCore limits

The **StandardCore** limit is a compute metric for the resources that are temporarily consumed when deploying the managed application.

It is possible that the Ansible Automation Platform for Azure will deploy without hitting the **StandardCore** limit. When a deployment fails because the consumed resources hit the **StandardCore** limit, the error message includes *container group quota 'StandardCores' exceeded*:

```markdown
<table>
<thead>
<tr>
<th>code</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeploymentFailed</td>
<td>At least one resource deployment operation failed. Please list deployment operations for details. Please see <a href="https://aka.ms/DeployOperations">https://aka.ms/DeployOperations</a> for usage details.</td>
</tr>
<tr>
<td>details</td>
<td>- code: DeploymentScriptContainerGroupInvalidSettings</td>
</tr>
</tbody>
</table>
```

**Requesting StandardCore limit increase**

The **StandardCore** metric is not displayed in the **My Quotas** page in the Azure console. To request the value of your regional limit, contact Microsoft directly.

If your deployments fail because the consumed resources reach this limit, you must submit a resource increase request for **StandardCore** to Microsoft. Only submit a quota increase request if you encounter a deployment failure due to this issue.

Use the following information to respond to questions from Microsoft support:

**Will the container groups be run in Linux or Windows?**

- **Linux**

**What will the core and memory be in your Container Group instance?**

- Red Hat recommends 20 cores, 16 GB

**When will you create all the Container Group Instances?**

- During managed application deployment of Red Hat Ansible Automation Platform on Microsoft Azure

**How frequent will you create/delete the container groups?**

- Only during managed application deployment of Red Hat Ansible Automation Platform on Microsoft Azure

**2.2. CREATING A SERVICE PRINCIPAL**

To allow the Ansible Automation Platform application to access and manage Azure resources, you must provide authorization credentials. The Microsoft Azure collection supports service principal authentication.
To create a service principal, you must have administrator privileges with tenancy-wide permissions on your Azure tenant. Your Ansible on Azure deployment will be provisioned in the same Subscription ID as the service principal created in this step.

1. Navigate to the Azure portal and click the Cloud Shell icon to open a bash Cloud Shell in your browser.

2. Set the Azure CLI to use the subscription that you intend to use for automating Azure services. Run the following command from the shell:

   ```
   az account set --subscription <your_subscription_id>
   ```

3. Run the following command using the Azure CLI to create a privileged service principal in Azure AD:

   ```
   az ad sp create-for-rbac --name ansible --role Contributor
   ```

   The output displays the `appId` and `tenant` keys for the service principal:

   ```
   {
     "appId": "xxxxxxx-xxx-xxxx",
     "displayName": "ansible",
     "name": "xxxxxxx-xxx-xxxx",
     "password": "xxxxxxx-xxx-xxxx",
     "tenant": "xxxxxxx-xxx-xxxx"
   }
   ```

4. Store the service principal details securely, as they are displayed only when you create the secret. You will need them when you deploy Automation controller.

### 2.2.1. Maintaining your service principals

Service principal credentials have a limited lifetime that is set in your Azure AD configuration. Track the lifespan of the service principal if you intend to automate against Azure for an extended period of time. You can create a new one when needed.

To view records of updated or deleted service principles, run the following Azure CLI command:

```
az ad sp list -o table | grep ansible
```

This command does not display the secrets for your service principals. Delete the service principal and create a new one if the secret is lost.

When you create a new service principal to replace an expired or deleted one, you must update the credential that uses the service principal that you are replacing. If the credential is not updated, automations that use that credential will fail.

### 2.3. DEPLOYING ANSIBLE AUTOMATION PLATFORM FROM AZURE MARKETPLACE

#### 2.3.1. Locating Ansible Automation Platform in Azure Marketplace

1. In a browser, navigate to the Azure Marketplace.
2. Click **Private Products** from the menu on the left of the screen.


4. Click the card that is returned in the search. Be sure to select the official offering from Red Hat.

5. Click **Get it Now**, **Continue**, and then **Create** to initiate the deployment process.

### 2.3.2. Provisioning Red Hat Ansible Automation Platform on Microsoft Azure

When you initiate the deployment of the Red Hat Ansible Automation Platform managed app from Azure marketplace, a form is displayed in the **Create Red Hat Ansible Automation Platform on Microsoft Azure** window. Complete the form to provision Red Hat Ansible Automation Platform infrastructure and resources into your Azure tenant.

1. Enter values for your deployment in the following fields in the form:
   - **Resource Group**: Create or select a resource group where you want to deploy the managed application.
   - **Region**: The Azure region where the application will be deployed.
   - **Automation Platform Cluster Name**: A unique name for the Kubernetes cluster.
   - **Administrator Password**: Create an administrator password for your deployment. The **Administrator Password** must contain at least 8 characters, and must include uppercase letters, lowercase letters, and numbers.
   - **Confirm Administrator Password**: Confirm the **Administrator Password**.
   - **Registry Username**: Your Red Hat SSO username. The **Registry Username** is the Red Hat account that will be used to pull the container images from registry.redhat.io.
   - **Registry Password**: Your Red Hat SSO password.
   - **Confirm Password**: Confirm your Red Hat SSO password.
   - **Application Name**: A unique name for the managed application.
   - **Managed Resource Group**: A resource group for the managed application infrastructure. Keep this resource groups isolated from other resource groups, including the **Resource Group** where you will deploy the managed application.

2. Store the information that you entered in the form in a secure place. You will need to provide the **Administrator password** to access automation controller and private automation hub.

3. Click **Review + Create**.

4. If the information you entered in the form is valid, the window displays **Validation Passed**.

5. Select **I agree** to accept the Co-Admin Access Permissions terms and conditions.

6. Click **Create** to begin the provisioning process for the application.

The application will begin provisioning.

It may take 30 minutes or longer for the infrastructure and software to fully provision.
Once provisioning is complete, you can access and login to your new Ansible Automation Platform instance and launch automation controller and automation hub.

### 2.4. ACCESSING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE

When you initiate the deployment of the Red Hat Ansible Automation Platform managed app from Azure marketplace, a form is displayed in the Create Red Hat Ansible Automation Platform on Microsoft Azure window. Complete the form to provision Red Hat Ansible Automation Platform infrastructure and resources into your Azure tenant.

1. In a web browser, navigate to **Managed Applications** in the Azure console.
2. Select the instance of Red Hat Ansible Automation Platform on Microsoft Azure that you deployed.
3. Select **Parameters and Outputs** in the **Settings** section in the left navigation menu.
4. Copy the URL links for automation controller and automation hub, and then save them. The names for the links are `automationControllerUrl` and `automationHubUrl`.
5. In a browser, navigate to the automation controller URL, and then log in using the following credentials:
   - **Username**: `admin`
   - **Password**: Use the *Administrator password* you provided when you deployed the AAP application.

The first time you login to Ansible Automation Platform, you must configure a subscription and agree to the terms and conditions.

#### 2.4.1. License association

Red Hat provided a specific subscription entitlement manifest when you subscribed to Red Hat Ansible Automation Platform on Microsoft Azure.

When asked to submit information about your license, select your license manifest file that you obtained from access.redhat.com.

#### 2.4.2. Azure Active Directory (Azure AD) SSO configuration

Follow the procedures below to configure SSO with Azure Active Directory (Azure AD). If your organization does not use Azure AD for application authorization, you can create users in the user management system in Ansible Automation Platform.

**Configuring the base URL for the Ansible Automation Platform deployment**

1. In a browser, navigate to the Automation controller URL and log in using the following credentials:
   - **Username**: `admin`
   - **Password**: Use the *Administrator password* you provided when you deployed the AAP application.
2. In the Automation controller console, click **Settings** in the menu options.

3. Click **Miscellaneous System settings** under the **System** settings.

4. Click **Edit**. Enter the Automation controller URL in the Base URL of the service field.

5. Click **Save**.

**Configuring authentication for Ansible Automation Platform**

To set up enterprise authentication for Microsoft Azure Active Directory (Azure AD), you must obtain an OAuth2 key and secret by registering your Ansible Automation Platform deployment in Azure.

To register the automation controller instance in Azure, you must supply the *Azure AD OAuth2 Callback URL* from the automation controller settings.

**Fetching the Azure AD OAuth2 Callback URL**

1. In a web browser, open the automation controller console.

2. Click **Settings** in the menu to open the main settings page.

3. Click **Azure AD settings** in the **Authentication** category to open the **Details** page.

4. Copy the value for **Azure AD OAuth2 Callback URL**. You will need this value when you register your deployed application in Azure AD.

**Creating a registered application in Azure AD**

1. In a web browser, open the Azure portal.

2. Ensure that you are using the tenant where you deployed Ansible Automation Platform.

3. Type **Azure Active Directory** in the search bar.

4. Select **Azure Active Directory** from the search results.

5. Under **Manage** in the menu options, click **App registrations**.

6. In the **App registrations** page, click **+ New registration**.

7. Configure the new registration as follows:
   
   - In the **Name** field, enter the same name that you used for the deployed application.
   - Select the default value for **Supported account types**.
   - Select **Web** for **Redirect URI (optional)**.
   - In the **Redirect URI (optional)** field, enter the *Azure AD OAuth2 Callback URL* value that you fetched from automation controller.

8. Click **Register** to create the registration.

When registration is complete, the registration page for the Automation Controller application is displayed.
Generating secrets for communication

1. In the **Automation controller application registration page** on Azure, copy and save the value of **Application (client) ID**.
   You will use this value for the **Azure AD OAuth2 Key** in the Ansible Automation Platform settings.

2. Under **Manage**, click **Certificates & secrets**

3. Click **Client secrets** and then **+ New client secret**

4. Provide a description for the new secret.
   It is not possible to automatically renew a certificate or identify when it is about to expire.
   It is useful to include the date in the description, for example: AAP Client Secret <Today’s Date in YYYY-MM-DD format>.

5. Provide an expiration date for the new secret.
   The maximum lifetime for the certificate is 2 years. Unless you have specific security needs that prevent the creation of a long term certificate, select an expiration date of **24 months**.

6. Save the secret Value to a location on your local machine. Once you navigate away from this page it will be hidden and no longer retrievable.

Adding secrets to Ansible Automation Platform settings

Add the key (**Application (client) ID**) and value (**Value**) of the secret that you generated in Azure to your Ansible Automation Platform instance.

1. Open the automation controller console in a web browser.

2. Click **Settings → Azure AD settings**.

3. Click **Edit**.

4. Enter the information for the secret that you generated in Azure AD:

5. In **Azure AD OAuth2 Key**, paste the **Application (client) ID**.

6. In **Azure AD OAuth2 Secret**, paste the secret **Value**.

7. Click **Save**.

Adding Azure Credentials to Automation controller

1. Open the automation controller in a web browser.

2. Under **Resources**, click **Credentials**.

3. Click **Add** to open the **Create New Credentials** page.

4. Enter a name for the new credential and select **Azure Resource Manager** for the credential type.

5. Use the Service Principal details to fill out the values of the form:
   - **Name**: Choose a descriptive name for the credential, for example *Azure Infrastructure*.
- **Subscription ID**: Enter the subscription id where your resources created in Azure should be associated. This is unique to your tenant. Your organization may have multiple subscription ids; consult your Azure administrator regarding the subscription id that you should use.

- **Client ID**: Enter the appId value from the Service Principal creation.

- **Client Secret**: Enter the password from the Service Principal creation.

- **Tenant ID**: Enter the tenant from the Service Principal creation.

6. Click **Save** to save the credential.
CHAPTER 3. PRIVATE NETWORK PEERING

Ansible Automation Platform on Azure is deployed into an independent managed resource group with its own Azure virtual network (VNet).

When initially deployed, Ansible Automation Platform on Azure’s VNet can only send requests to external networks through the public internet.

You can configure your Azure VNets to allow private communication between multiple Azure VNets as well as private transit routing between Azure VNets and external VPN routed networks. These VPN networks can be on-premises or on other clouds.

NOTE

While every effort has been made to align with Microsoft’s documentation for this content, there may be drift in accuracy over time. Microsoft’s documentation is the definitive source for information about networking topics for Azure.

Azure offers different ways to peer private networks. These are typically divided into two categories:

- **Hub-and-spoke peering**: In this topology, there is a centralized hub VNet that other virtual networks peer with. This hub network has mechanisms to route traffic through transit routing. Cloud networks, including VPN/Express Connect connections with on-premises and other cloud networks, can communicate through the hub VNet.

- **Direct peering**: Private networks are individually connected to one another with no routing hops between them. This is a simpler peering model: it is useful when you only want to connect a few networks.

3.1. HUB-AND-SPOKE PEERING (TRANSIT ROUTES)

NOTE

Updating route tables incorrectly can break your network. Only execute the steps in these procedures if you are confident that you can reverse any unexpected network behavior.

3.1.1. Hub-and-spoke peering process overview

Prerequisites

- You have deployed Ansible Automation Platform on Microsoft Azure.

- You have configured and tested an Azure VNet hub-and-spoke implementation in your Azure tenant.

- You have configured transit routing between your spoke networks, including your VPNs. Refer to Configure VPN gateway transit for virtual network peering in the Microsoft Azure documentation for instructions.

- You have identified the following:
  - The CIDR block(s) of your existing VNets (including VPNs & direct connects) that will need access to Ansible Automation Platform on Azure UIs.
- The CIDR block(s) of your existing VNets (including VPNs & direct connects) that will contain hosts or endpoints for Ansible automation.

- The CIDR blocks of the Ansible Automation Platform Kubernetes network from the managed resource group of the application. Refer to Finding the CIDR Block of the managed application Kubernetes cluster for instructions.

Before peering any networks, ensure that there is no network address space overlap between your private VNets and your Ansible Automation Platform on Azure network.

Procedure

1. Find the CIDR Block for the Ansible Automation Platform on Microsoft Azure managed application Kubernetes cluster. See Finding the CIDR Block of the managed application Kubernetes cluster.


3. Update the route tables:
   
a. Configure route tables from your existing networks to send traffic to the managed application CIDR. You must add routes to the routing tables of every network requesting Ansible Automation Platform user interfaces and of every network that will have automation performed against its resources. See Routing to Ansible Automation Platform on Microsoft Azure.

b. Configure routing to your VNets for each spoke network that you would like PlatformNameShortAnsible Automation Platform to communicate with, for automation or for accessing the user interfaces. See Routing to your VNets.

3.1.1.1. Finding the CIDR Block of the managed application Kubernetes cluster

1. Open the Virtual Networks page in the Azure portal to list your VNets. The VNet name for the AKS cluster of your managed application deployment follows the format: aks-vnet-<digits>.

2. Select a VNet to view its settings in the Overview page. The CIDR block of the cluster is displayed in the Address Space.

If there are multiple VNet names with the aks-vnet-<digits> format, open the Overview page for each VNet until you find the network within your Ansible Automation Platform on Microsoft Azure managed application resource group.

For further information, refer to View virtual networks and settings in the Microsoft Azure Virtual network guide.

3.1.1.2. Configuring network peering with the Ansible Automation Platform subnet

Within the Azure console, the Azure virtual network (VNet) is known as this virtual network, and the VNet that you want to peer with is known as remote virtual network.

In the Virtual Networks page in the Azure portal, use the following settings to configure peering between the Azure VNet and the VNet that you want to peer with the Ansible Automation Platform on Microsoft Azure app:
• Under **This virtual network**, select settings for the Ansible Automation Platform on Microsoft Azure virtual network:
  
  - **Peering link name** `<hub_to_aap_peering_link_name>`
  - **Traffic to remote virtual network** *Allow*
  - **Traffic forwarded from remote virtual network** *Allow*
  - **Virtual network gateway or Route Server** *Use this network’s gateway or Route server*

• Under **Remote virtual network**, select settings for the virtual network that you want to peer with Azure:
  
  - **Peering link name** `<aap_to_hub_peering_link_name>`
  - **Traffic to remote virtual network** *Allow*
  - **Traffic forwarded from remote virtual network** *Allow*
  - **Virtual network gateway or Route Server** *Use the remote virtual network’s gateway or Route server*

For further information on configuring peering, refer to Create a peering in the Microsoft Azure Virtual network guide.

### 3.1.1.3. Updating the route tables

Before you update the route tables, confirm that you satisfy the prerequisites for the hub-and-spoke peering process.

**Routing to Ansible Automation Platform on Microsoft Azure**

1. Navigate to Route Tables in the Azure portal.

2. As part of your hub-and-spoke configuration, you created one or more route tables to define the routes between the networks. Click on one of these route tables.

3. From the route table menu bar, click Routes > Add.

4. Configure routes from your existing networks to send traffic to Ansible Automation Platform. You must configure routes for any network requesting Ansible Automation Platform user interfaces and for any network that will have automation performed against its resources. For each route that you add, enter the following information:
   
   - **Route name**: Enter a route name for the Ansible Automation Platform managed application network
   - **Address Prefix**: The CIDR block of the managed application kubernetes cluster
   - **Next Hop Type**: Virtual network gateway

5. Click OK to save the new route to the route list.

Repeat this procedure for all other route tables where you want to route traffic to Ansible Automation Platform.

**Routing to your VNets**
Routing to your VNets

Add a route for each spoke network that you would like Ansible Automation Platform to communicate with, for automation or for accessing the user interfaces.

1. Navigate to **Route Tables** in the Azure portal.

2. In the list of route tables, select the route table for the Ansible Automation Platform on Microsoft Azure managed application.
   The name of the Ansible Automation Platform route table uses the following convention:

   aks-agentpool-<numbers>-routetable

3. From the route table menu bar, click **Routes > Add**.

4. Configure routing to your VNets for each spoke network that you would like Ansible Automation Platform to communicate with, for both automation or accessing the user interfaces.

5. For each route that you add, enter the following information:
   - **Route name**: Enter a route name for the spoke network that you want Ansible Automation Platform to route to
   - **Address Prefix**: The CIDR block of the spoke network
   - **Next Hop Type**: Virtual network gateway

6. Click **OK** to save the new route to the route list.

Once you have configured the routing rules, traffic is routed to and from Ansible Automation Platform on Azure entirely though your private network.

For further information about adding routes to a route table in Azure, refer to [Create a route](#) in the Microsoft Azure Virtual network guide.

### 3.2. DIRECT PEERING

You can use direct peering to directly connect virtual networks. When two networks are peered, Azure updates routes between them so that traffic automatically flows between them.

The direct peering method is easier to configure than the hub-and-spoke model. However, the number of direct network peerings is limited. Direct peering becomes difficult to manage as the number of virtual networks grows.

#### 3.2.1. Configuring direct network peering

You can configure network peering between your Azure network and your VNet in the Virtual Networks page of the Azure Portal.

Within the Azure console, the Azure virtual network is known as *this virtual network*, and the VNet that you want to peer with is known as *remote virtual network*.

In the Virtual Networks page in the Azure portal, use the following settings to configure the Azure network and the VNet that you want to peer with the Ansible Automation Platform on Microsoft Azure app:
• Under **This virtual network**, select settings for the Ansible Automation Platform on Microsoft Azure virtual network:
  - **Peering link name** `<hub_to_aap_peering_link_name>`
  - **Traffic** to remote virtual network: *Allow*
  - **Traffic** forwarded from remote virtual network: *Allow*
  - **Virtual network gateway or Route Server** *Use this network’s gateway or Route server*

• Under **Remote virtual network**, select settings for the virtual network that you want to peer with Azure:
  - **Peering link name** `<aap_to_hub_peering_link_name>`
  - **Traffic to remote virtual network** *Allow*
  - **Traffic forwarded from remote virtual network** *Allow*
  - **Virtual network gateway or Route Server** *Use the remote virtual network’s gateway or Route server*

For more detailed instructions for configuring peering, refer to [Create a peering](#) in the Microsoft Azure Virtual network guide.

For further information on direct peering, refer to [Virtual network peering](#) in the Microsoft Azure Virtual network guide.