



# Red Hat Ansible Automation Platform 2.1

## Red Hat Ansible Automation Platform on Microsoft Azure guide

Install and configure Red Hat Ansible Automation Platform on Microsoft Azure



# Red Hat Ansible Automation Platform 2.1 Red Hat Ansible Automation Platform on Microsoft Azure guide

---

Install and configure Red Hat Ansible Automation Platform on Microsoft Azure

## Legal Notice

Copyright © 2023 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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, please contact technical support at to create an issue on the Ansible Automation Platform Jira project using the Docs component.

# Table of Contents

<b>PREFACE</b> .....	<b>3</b>
<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>4</b>
<b>CHAPTER 1. INTRODUCTION TO ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE</b> .....	<b>5</b>
1.1. ABOUT RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE	5
1.2. APPLICATION ARCHITECTURE	5
1.2.1. Public deployment	6
1.2.2. Private deployment	7
1.3. NETWORK	8
1.4. ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE INFRASTRUCTURE USAGE	9
1.5. LIFECYCLE MANAGEMENT	10
1.6. ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE SCALING	10
1.7. MIGRATION	10
<b>CHAPTER 2. INSTALLING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE</b> .....	<b>11</b>
2.1. PREREQUISITES	11
2.1.1. Azure resource quotas and infrastructure limits	11
2.1.1.1. Regional vCPU limits	11
2.1.1.2. Regional StandardCore limits	11
2.2. CREATING A SERVICE PRINCIPAL	12
2.2.1. Maintaining your service principals	13
2.3. DEPLOYING ANSIBLE AUTOMATION PLATFORM FROM AZURE MARKETPLACE	13
2.3.1. Locating Ansible Automation Platform in Azure Marketplace	13
2.3.2. Provisioning Red Hat Ansible Automation Platform on Microsoft Azure	14
2.4. ACCESSING RED HAT ANSIBLE AUTOMATION PLATFORM ON MICROSOFT AZURE	15
2.4.1. License association	15
2.4.2. Azure Active Directory (Azure AD) SSO configuration	15
<b>CHAPTER 3. PRIVATE NETWORK PEERING</b> .....	<b>19</b>
3.1. HUB-AND-SPOKE PEERING (TRANSIT ROUTES)	19
3.1.1. Hub-and-spoke peering process overview	20
3.1.1.1. Finding the CIDR Block of the managed resource group	20
3.1.1.2. Configuring network peering with the Ansible Automation Platform subnet	21
3.1.1.3. Updating the route tables	21
3.1.1.3.1. Additional resources	23
3.2. AZURE VIRTUAL WAN (VWAN)	23
3.2.1. Peering a VWAN Hub to the Ansible Automation Platform on Microsoft Azure Network	23
3.3. DIRECT PEERING	24
3.3.1. Configuring direct network peering	24
<b>CHAPTER 4. CONNECTING TO RED HAT ANSIBLE AUTOMATION PLATFORM</b> .....	<b>26</b>
4.1. ACCESS DETAILS	26
4.2. PUBLIC DEPLOYMENTS	26
4.3. PRIVATE DEPLOYMENTS	26
4.3.1. Azure hosted virtual machine	26
4.3.2. VPN	27
4.3.3. SSH tunnel	28



## PREFACE

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. This document has been updated to include information for the latest release of Ansible Automation Platform on Microsoft Azure.

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).



# 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 Red Hat Ansible Automation Platform on Microsoft 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 Red Hat Ansible Automation Platform



### NOTE

Automation mesh is not available on Ansible Automation Platform on Microsoft 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 the following resource groups:

- A new or existing resource group (RG) in your tenant. This resource group includes a single resource referring to the Ansible Automation Platform on Microsoft Azure managed application deployment. Red Hat has access to the managed app to perform support, maintenance, and upgrades, but the resource group is outside of Red Hat's management.

- A multi-tenant managed resource group (MRG) that contains most of 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.
- An AKS node pool resource group (NPRG). Microsoft requires the NPRG for AKS deployments. It contains resources that AKS uses to function. It is created on deployment, and it is outside of Red Hat's management. Refer to [Microsoft's AKS documentation](#) for more information about NPRGs.



## NOTE

Do not interact with any resources in the node pool resource group (NPRG) unless explicitly directed to by the Red Hat Ansible Automation Platform on Microsoft Azure SRE team. Changes to resources in the NPRG cannot be protected by Red Hat and can cause irrecoverable damage to the application.

Red Hat cannot restrict your ability to change or delete resources in the NPRG.

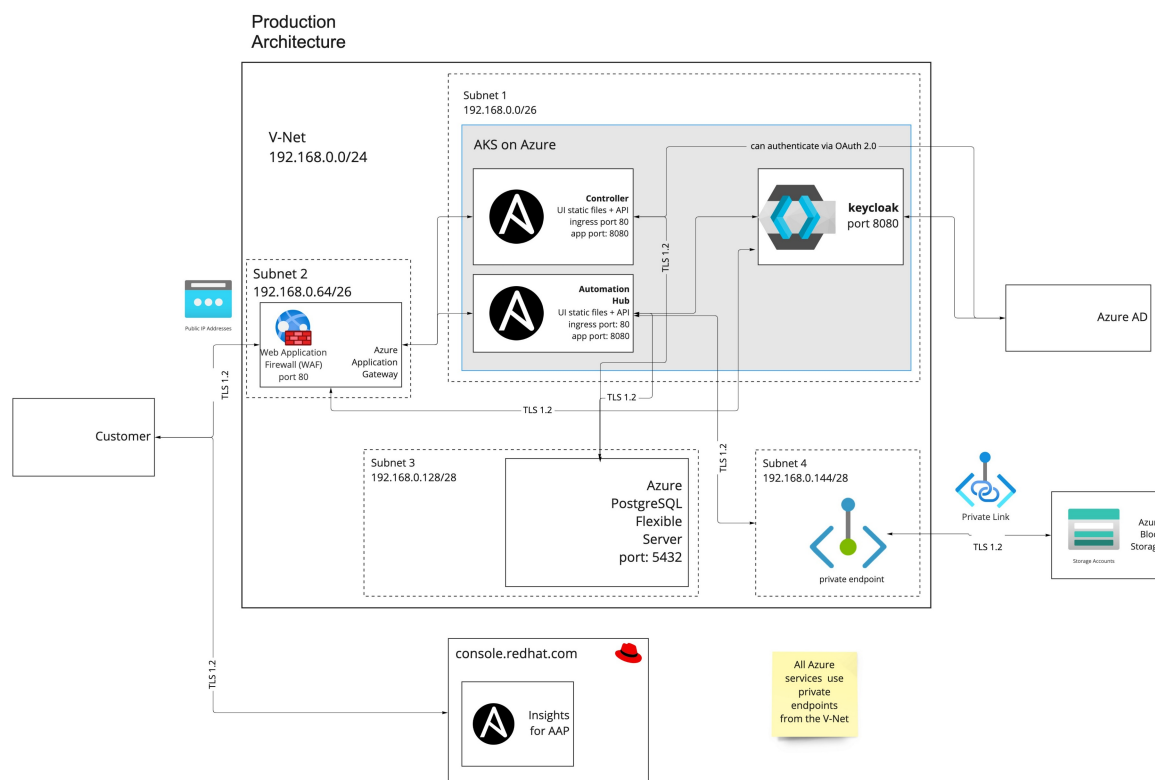
When you install Ansible Automation Platform on Microsoft Azure, you choose whether the deployment is public or private. This affects how users can access the Ansible Automation Platform user interfaces.

Regardless of whether you choose a public or private deployment, you must configure network peering for outbound communication from Ansible Automation Platform to the private networks that contain resources that you want to automate against. You can configure network peering from Ansible Automation Platform on Microsoft Azure to your private Azure VNets and to on-premises or multi-cloud networks where transit routing with Azure exists.

### 1.2.1. Public deployment

Public deployments permit ingress to the Ansible Automation Platform on Microsoft Azure user interfaces over the public internet. Upon deployment, a domain name is issued to the Ansible Automation Platform on Microsoft Azure instance. No configuration is required to access Ansible Automation Platform. Users can navigate to the domain from the public internet and log in to the user interfaces.

The following diagram outlines the application resources and architecture that are deployed into the managed application resource group on a public deployment of Ansible Automation Platform on Microsoft Azure into your Azure subscription. The IP ranges change based on the networking address range you set on deployment.



## 1.2.2. Private deployment

A private deployment of Ansible Automation Platform resides in an isolated Azure VNet with no access from external sources: traffic to and from the public internet and other Azure VNets and subnets is blocked.

To access the URLs for the Ansible Automation Platform user interfaces, you must configure network peering.

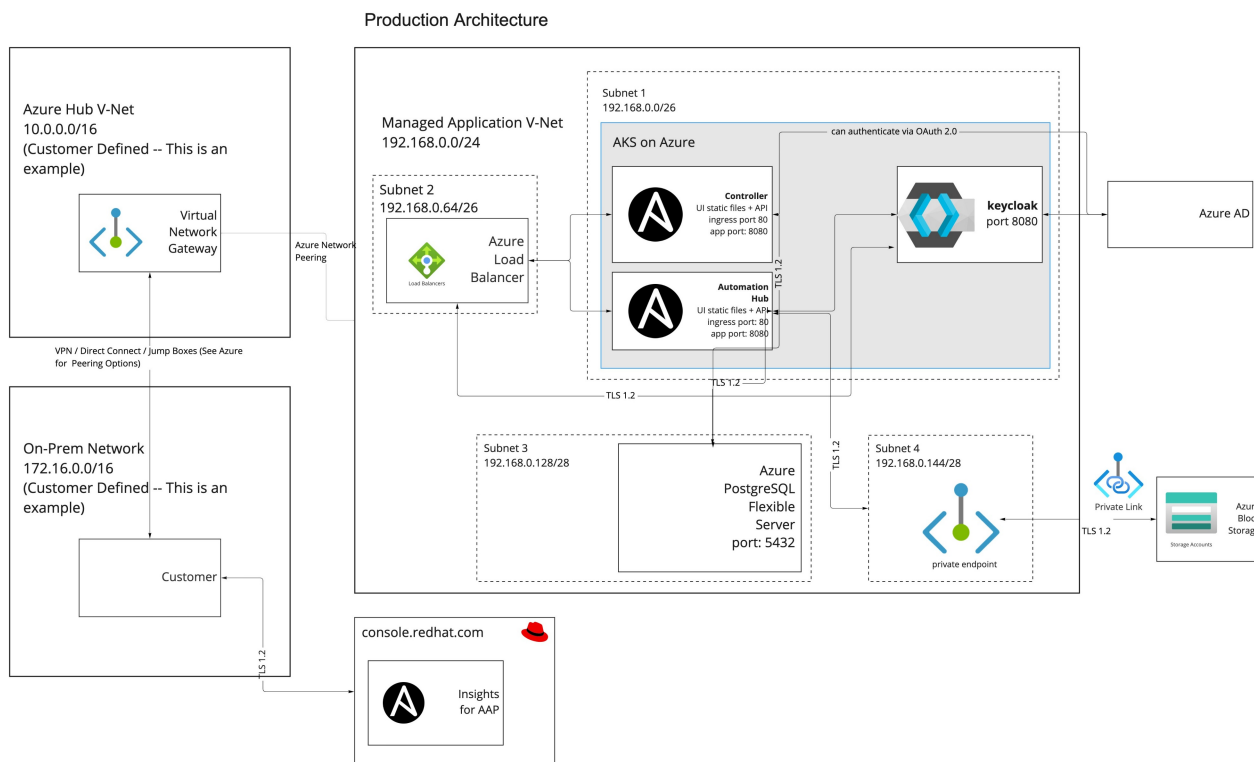
Once peering and routing are configured, users can access Ansible Automation Platform through a VM on a connected Azure subnet, or directly if your organization has transit routing set up between Azure and your local network.



### NOTE

No two Azure networking configurations are the same. To allow user access to your Ansible Automation Platform URLs, your organization will need to work with your Azure administrators to connect the private access deployment.

The following diagram outlines the application resources and architecture that are deployed into the managed application resource group on a private deployment of Ansible Automation Platform on Microsoft Azure into your Azure subscription. The IP ranges change based on the networking address range you set on deployment.



### 1.3. NETWORK

You can configure the networking address range (CIDR block) for the VNet that your Ansible Automation Platform on Microsoft Azure application uses. You set the CIDR block for the application in a network configuration form when you deploy Ansible Automation Platform on Microsoft Azure. Plan your networking configuration before you deploy the Ansible Automation Platform on Microsoft Azure application, because you cannot change it after deployment.

When you are planning your network configuration, bear the following in mind:

- The managed application requires at least a /24 Vnet that is divided into four subnets. The subnets have minimum address spacing.

Networking entity	Minimum CIDR Block
VNet	/24
Cluster subnet	/26
Gateway subnet	/28
Database subnet	/28
Private link subnet	/28

- Ensure that the VNet range you configure does not intersect with the default CIDR block for AKS clusters (10.0.0.0/16). The Azure user interface does not prevent you entering this range, but using the default AKS CIDR block for your VNet causes networking issues.
- To ensure successful network peering and communication between Ansible Automation Platform on Microsoft Azure and your existing networks, your enterprise network ranges must not overlap with the VNet network range.
- If you do not have any existing Azure VNets, the Azure user interface suggests a default CIDR block and range for the VNet. Do not accept these defaults. Instead, use the network configuration that you have planned.

For information about planning the network address range and completing the networking configuration form on deployment, refer to [Red Hat Ansible Automation Platform on Microsoft Azure VNet Preparation](#).

## 1.4. 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:

### Azure Kubernetes service (AKS)

The Kubernetes cluster used to deploy Ansible Automation Platform applications and services.

#### Service Shape:

- Compute nodes: Standard\_D4s\_v3 (4 vCPUs x 16 GiB)
- Autoscaling minimum nodes: 2
- Autoscaling maximum nodes: 20

### Managed identity

An Azure service that enables Ansible Automation Platform components to communicate with other Azure services such as database, DNS, storage, and other services.

### Key vault

A secure key vault used to store secrets that are unique to the Ansible Automation Platform deployment.

### Log Analytics Workspace

An Azure service that enables Red Hat site reliability engineers to inspect the operations of Ansible Automation Platform on Microsoft Azure.

### Private DNS Zone

Manages local DNS requests for the services used by Ansible Automation Platform on Microsoft Azure.

### Azure Database for PostgreSQL

An Azure database service used for Ansible Automation Platform's PostgreSQL database.

**Service Shape:** 1 TB

### Storage account

The Azure service is used for file and block storage for local storage of projects, containers, etc.

**Service Shape:** StorageV2 - Standard\_LRS

## Virtual network

The Azure service is used to manage all internal networking and dependent services such as the Azure Application Gateway.

**Service Shape:** Application Gateway: WAF\_v2

Exact infrastructure usage depends on the length of time that the managed application is deployed in your tenancy, and the automation requirements that might cause the Kubernetes cluster to autoscale to meet the demands of your workload.

Microsoft provides a [Pricing calculator](#) to estimate your costs for Azure products and services. Red Hat has configured an example scenario in the pricing calculator: use the [Red Hat Ansible Automation Platform on Azure Infrastructure Estimate](#) to tune Kubernetes expected auto scaling variables based on your organization's workloads.

## 1.5. LIFECYCLE MANAGEMENT

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

## 1.6. 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

## 1.7. 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](https://access.redhat.com)).
- A specific subscription entitlement for Red Hat Ansible Automation Platform.

### 2.1.1. 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 on Microsoft Azure, ensure that you have capacity to deploy the managed application into your desired region. Refer to [Azure infrastructure usage](#) for infrastructure requirements.

#### 2.1.1.1. 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 80 DS2\_V3 vCPUs available 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.

#### 2.1.1.2. 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 on Microsoft Azure can 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**:

```
code: DeploymentFailed
message:
  At least one resource deployment operation failed. Please list deployment operations for details.
  Please see https://aka.ms/DeployOperations for usage details.
details:
- code: DeploymentScriptContainerGroupInvalidSettings
  message:
    Resource type 'Microsoft.ContainerInstance/containerGroups'
    container group quota 'StandardCores' exceeded in region 'eastus'.
    Limit: '10', Usage: '10' Requested: '1'.
```

### 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 enable the Ansible Automation Platform application to access and manage Azure resources, you must provide authorization credentials after deployment. 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 Automation Platform on Microsoft 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. From the navigation panel, select **Private Products** from the menu on the left of the screen.
3. Search for Red Hat Ansible Automation Platform.
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.

Before you fill in the form, decide whether you want to create a public or private deployment of Ansible Automation Platform on Microsoft Azure:

- Public deployments allow ingress to the Ansible Automation Platform on Microsoft Azure user interfaces over the public internet. No configuration is required to access the application URLs.
- Private deployments are created in an isolated Azure VNet that blocks access from the public internet. To access Ansible Automation Platform on Microsoft Azure user interfaces, you must configure network peering and routing.

You create the network configuration for the Ansible Automation Platform on Microsoft Azure VNet when you initiate the deployment. Refer to your network configuration plan before deploying the managed application. For information about planning your network configuration, see [Networking configuration](#).

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:
  - **Subscription:** Select **Ansible on Clouds**.
  - **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.
  - **Application Name:** A unique name for the managed application.
  - **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*.
  - **Access:** Choose whether your deployment will be public or private.
  - **Managed Resource Group:** A resource group for the managed application infrastructure. Keep this resource group 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 **Next**
4. Follow the steps in [Red Hat Ansible Automation Platform on Microsoft Azure VNet Preparation](#) to configure your network configuration.
5. Click **Review + Create**.
6. If the information you entered in the form is valid, the window displays **Validation Passed**.

7. Select **I agree** to accept the Co-Admin Access Permissions terms and conditions.
8. 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 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 Ansible Automation Platform application.

The first time you login to Ansible Automation Platform on Microsoft Azure, 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](https://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 Ansible Automation Platform application.
2. In the Automation controller console, click **Settings**.
3. Click **Miscellaneous System settings** under the **System** settings.
4. Click **Edit**. Enter the Automation controller URL in the Base URI 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. From the menu options, select **Manage** → **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. Select **Manage** → **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. Select **Resources** → **Credentials** and click **Add** to open the **Create New Credentials** page.

3. Enter a name for the new credential and select **Azure Resource Manager** for the credential type.
4. 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.
5. Click **Save** to save the credential.

## CHAPTER 3. PRIVATE NETWORK PEERING

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

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

To enable Ansible Automation Platform on Microsoft Azure to access resources in an internet-gapped deployment, when access to resources has to happen over private networks, you must configure Azure network peering between your private virtual networks and Red Hat Ansible Automation Platform on Microsoft Azure's managed application VNet.

You can configure your Azure VNets to enable 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.

No two Azure networking configurations are the same. To enable user access to Ansible Automation Platform on Microsoft Azure, work with your Azure administrators to connect your deployment to your VNets and external VPN routed networks.



### NOTE

Network peering must be configured by Azure administrators in your organization who are familiar with Azure networking. Configuring network changes to your Azure account can cause outages or other disruptions.

The network peering procedures described in this document are not supported by Red Hat, as the processes and services are controlled and managed by Microsoft Azure. Contact Microsoft for assistance in peering Azure networks.

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.
- **Azure Virtual WAN (VWAN)** Azure Virtual WAN is a networking service that provides simplified hub-and-spoke network modeling across Azure, on-premises, and other VPN/Direct Connect networks. For more about VWAN, refer to Microsoft's [Virtual WAN documentation](#).
- **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.

Refer to [Choose between virtual network peering and VPN gateways](#) in the Microsoft *Application architecture fundamentals guide* to determine the correct peering approach for your organization.

### 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. This prerequisite requires many Azure resources to be configured, including a Virtual Network Gateway.
- 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 Microsoft Azure Uls.
  - 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 on Microsoft Azure VNet from the managed resource group of the application. Refer to [Finding the CIDR Block of the managed resource group](#) 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 Microsoft 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](#).
2. Configure Network Peering with the Ansible Automation Platform Subnet. See [Configuring Network Peering with the Ansible Automation Platform Subnet](#).
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 Ansible 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 resource group



1. Navigate to the **Resource Groups** page in the Azure portal.
2. Click the managed resource group for Red Hat Ansible Automation Platform on Microsoft Azure. The resource group name is prefixed with “-mrg”.
3. Select the VNet within the resource group to view its settings in the **Overview** page. The CIDR block of the cluster is displayed in the **Address Space**.

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, navigate to **Routes → Add**.
4. From the route table menu bar, select **Routes → Add**.
5. 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*
6. 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

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, navigate to **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.

After you have configured the routing rules, traffic is routed to and from Ansible Automation Platform on Azure through your hub network.

### Outbound routing through virtual appliances

If your organization uses Azure firewall services or third-party firewall appliances through a Virtual Appliance connection, you must configure outbound connectivity from the managed application, to enable Red Hat to maintain your application and to enable automation against external resources.

The easiest way to implement this is to create a firewall rule that allows all outbound traffic from port 443.

If you choose not to allow all outbound traffic from port 443, you must configure routes.

- For Red Hat to manage and upgrade Ansible Automation Platform on Microsoft Azure and execute security patching, any machine in the Azure Kubernetes service (AKS) cluster must be allowed to submit a request to pull updates for containers used by Ansible Automation Platform. Add routes in the Ansible Automation Platform route table for outbound traffic from the full CIDR range of the Ansible Automation Platform on Microsoft Azure managed application to the following domains:
  - **registry.redhat.io**
  - **quay.io**
- You must also allow traffic from your firewall to any other external domain or IP address that you want Ansible Automation Platform to run automation jobs against. Otherwise, your firewall will block connectivity between Ansible Automation Platform and destinations for automation.

#### 3.1.1.3.1. Additional resources

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. AZURE VIRTUAL WAN (VWAN)

### 3.2.1. Peering a VWAN Hub to the Ansible Automation Platform on Microsoft Azure Network

Before peering, you must connect a hub network, and at least one spoke network, to the hub network of the Azure VWAN to which you intend to connect Ansible Automation Platform on Microsoft Azure.

#### Prerequisites

- A pre-configured Azure VWAN.
- One or more of the following connections to the VWAN:
  - A DMZ network that contains Azure virtual machines that users can remotely log into to access Ansible Automation Platform on Microsoft Azure.
  - A DMZ network that contains an Azure virtual machine that local machines can connect to with SSH tunneling to access Ansible Automation Platform on Microsoft Azure.
  - A VPN or Direct Connect service to your local network that routes traffic from local machines to Ansible Automation Platform on Microsoft Azure.

#### Procedure

1. Navigate to the **Virtual Network Connections** page for the VWAN that you want to peer with your Ansible Automation Platform instance.
2. To create a connection between the VWAN hub and your Ansible Automation Platform instance, use the following settings:
  - **Connection Name:** <Ansible\_Automation\_Platform\_connection\_name>
  - **Hubs:** Select one or more VWAN hub networks that the managed application VNet will peer with.
  - **Subscription:** Select the subscription where Ansible Automation Platform on Microsoft Azure has been deployed.
  - **Resource group:** The managed resource group of the managed application. It is typically prefixed with "mrg-".
  - **Virtual network:** The VNet of the managed application. There is only one VNet in the managed resource group.
  - **Propagate to none:** No
  - **Associate Route Table:** Select the default route table or the appropriate route table that your organization has configured for VWAN.
  - **Propagate to Route Tables:** Select one or more default route tables or the appropriate route table that your organization has configured for VWAN.
  - **Propagate to labels:** Select labels if your organization uses them.
  - **Static routes:** Do not complete this field.

When network peering completes, traffic routes to and from Ansible Automation Platform on Microsoft Azure through your VWAN hub network.

#### Additional resources

- [Connect a virtual network to a Virtual WAN hub - portal](#) (Microsoft Azure Virtual WAN documentation)

## 3.3. 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, because each new network requires peering to all other networks.

### 3.3.1. 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>*
  - **Subscription:** Select the subscription where Ansible Automation Platform on Microsoft Azure has been deployed
  - **Virtual network:** Select the Ansible Automation Platform on Microsoft Azure virtual network: *vnet-<aap\_identifier>-<region>*
  - **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*

After you have configured direct network peering, traffic routes between Ansible Automation Platform on Microsoft Azure and private hosts and IPs on your Vnet.

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.

## CHAPTER 4. CONNECTING TO RED HAT ANSIBLE AUTOMATION PLATFORM

When network peering is complete and your Azure routing configuration is established, you can choose how your team accesses Ansible Automation Platform through your Azure network configuration.

### 4.1. ACCESS DETAILS

Regardless of whether Ansible Automation Platform was deployed with public or private access, a set of DNS records is created. The DNS records are created so that Ansible Automation Platform can be issued a valid TLS certificate for your deployment and to enable easy access to your applications.

To view a list of the URLs for the Ansible Automation Platform applications, navigate to the **Parameters and Outputs** page of the Azure Marketplace managed application listing for your deployment.

### 4.2. PUBLIC DEPLOYMENTS

If you selected public access when you deployed Ansible Automation Platform on Microsoft Azure, you can access the Ansible Automation Platform application URLs over the public internet from a browser.

### 4.3. PRIVATE DEPLOYMENTS

If you selected private access when deploying Ansible Automation Platform on Microsoft Azure, then the DNS record issued to the Red Hat Ansible Automation Platform on Microsoft Azure application points to a private address within the CIDR block selected when the managed application was deployed. You must configure access to this address after you have created network peering.

The configuration and access method that you choose to connect to Ansible Automation Platform on Microsoft Azure depends on how your organization manages Azure infrastructure. Your Azure administrators must determine the right model for your organization and configure the setup for you.

The following are the most common options:

- [Azure hosted virtual machine](#)
- [VPN](#)
- [SSH tunnel](#)

#### 4.3.1. Azure hosted virtual machine

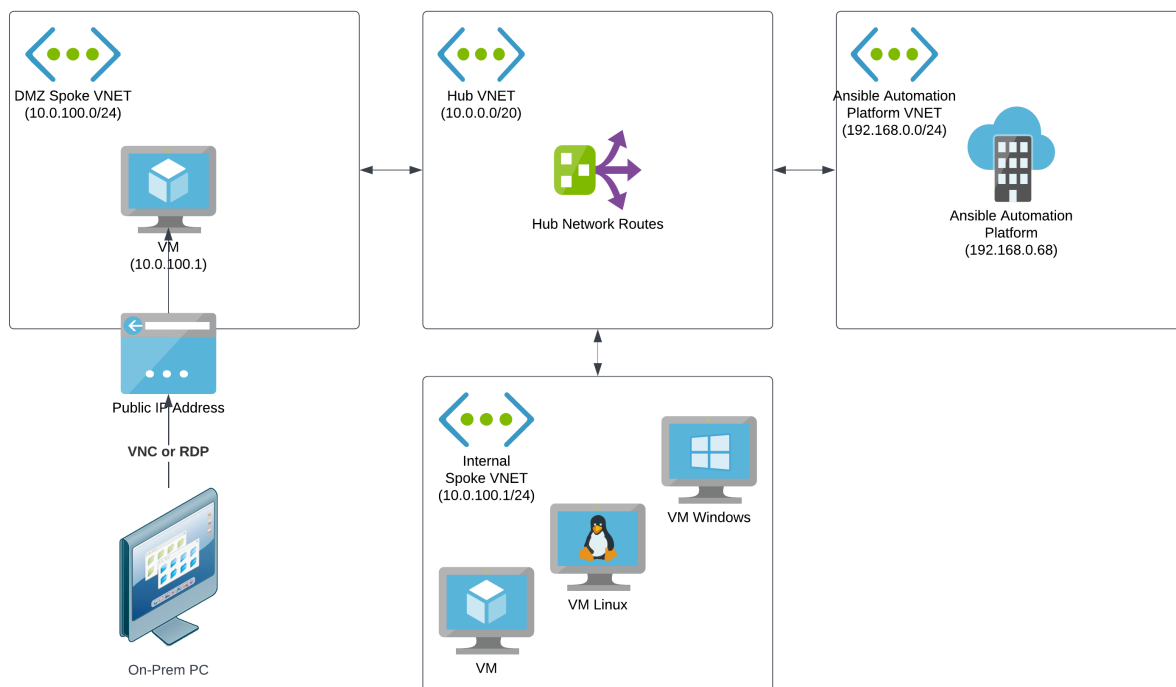
A straightforward way to configure access for a small set of users to access private network resources on Azure networks is to create a jumpbox VM in a perimeter network (DMZ VNet) that users can remotely log into from the public internet. The jumpbox VM requires workstation features such as a GUI and browser.

Users can remotely log into the publicly accessible virtual machine from on-premises machines through VNC, RDP, or other screen-sharing protocols.

To access the Ansible Automation Platform web UIs on the Azure private network, users navigate to the URLs using the browser on the jumpbox VM.

The DMZ VNet is connected to other Azure VNets through network peering, with routing rules established to send network traffic to the Ansible Automation Platform VNet.

The following diagram shows the topology for an example configuration of private network access via an Azure virtual machine.



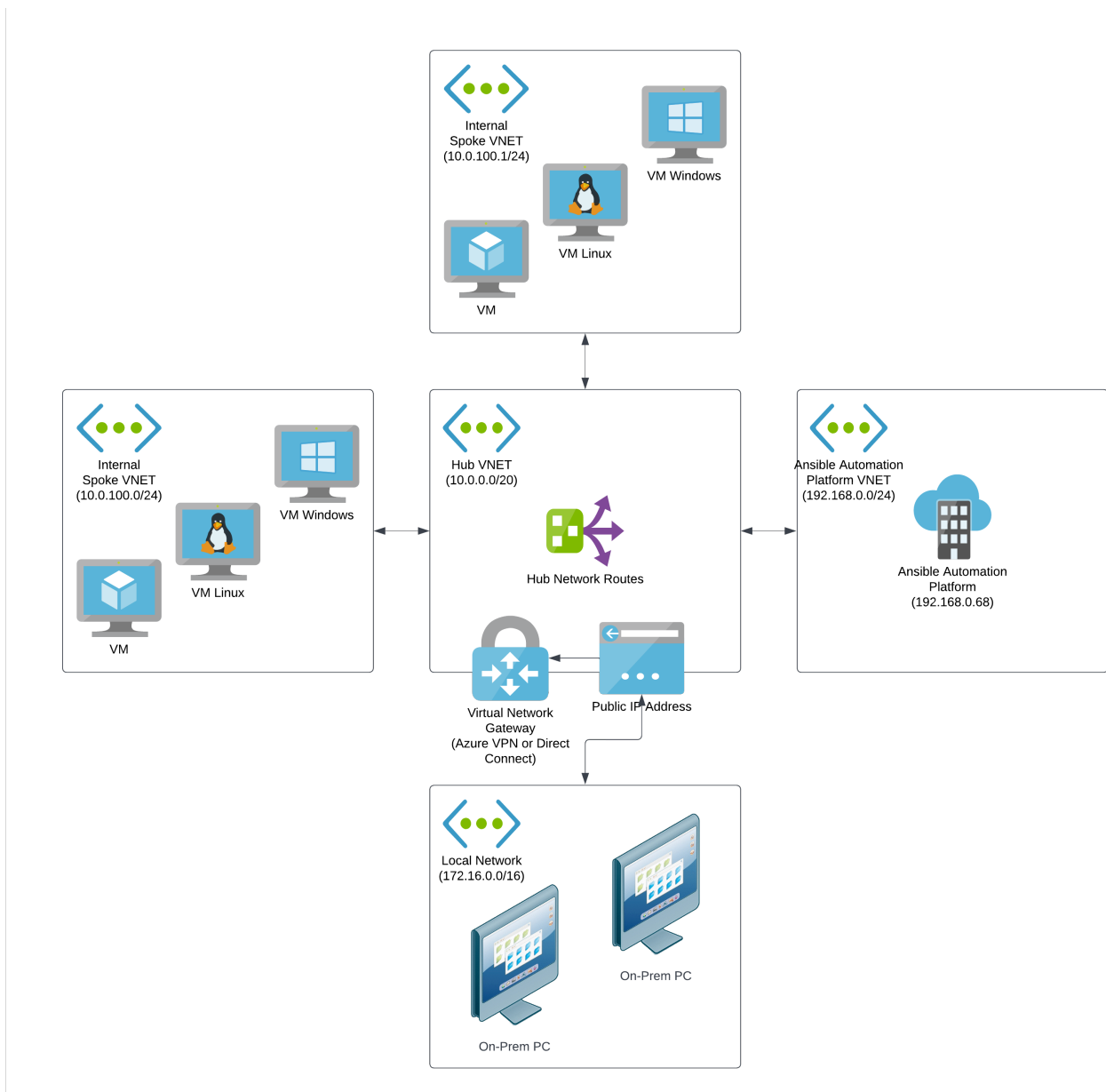
- For more information about perimeter (DMZ) networks, refer to [Perimeter Networks](#) in the *Microsoft Azure Cloud Adoption Framework* documentation.
- For more information about jumpboxes, refer to [About Azure bastion host and jumpboxes](#) in the *Microsoft Azure Cloud Adoption Framework* documentation.

### 4.3.2. VPN

If your organization requires that many users access Ansible Automation Platform over a private connection, or if your organization already uses VPNs or direct connections with Azure, then this approach might be suitable.

In this configuration, your on-premises infrastructure is connected to Azure through a Network Application Gateway and has routing rules that can enable access to Ansible Automation Platform to any connected computer on the local network. The VNet connected to the Virtual Network Gateway is connected to other Azure VNets through network peering, with routing rules established to send network traffic to the Ansible Automation Platform VNet.

With this configuration, users can access Ansible Automation Platform through the application URLs as if they were using the public access approach.



### 4.3.3. SSH tunnel

When VPN is not an option and your local users are more technical, the SSH tunnel approach is a secure alternative that enables users to access Ansible Automation Platform from a browser on a local machine.

To implement this access model, create a lightweight Linux-based SSH server in a DMZ network, similar to the Azure Hosted Virtual Machine method. The SSH server does not require any workstation features, because it simply acts as a proxy between a user's local machine and Ansible Automation Platform on Microsoft Azure.

Each user must be configured as an SSH user on the server. Users can then establish an SSH tunnel from the local machine to the SSH server to route traffic for Ansible Automation Platform on Microsoft Azure.

This approach is easier to implement on Linux and macOS host machines, but can be accomplished on Windows.

1. Update your local hosts file so that the Ansible Automation Platform URLs route traffic to your local machine rather than the private IP that DNS records are configured with. Add the following line to your hosts file:

```
127.0.0.1 controller.<your_AAPonAzure_instance>.az.ansiblecloud.com
```



The following example shows the line in a hosts file:

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
#
127.0.0.1    localhost
255.255.255.255 broadcasthost
::1        localhost

127.0.0.1 controller.<your_AAPonAzure_instance>.az.ansiblecloud.com
```

- As a user with root privileges, run the **ssh** command to establish an SSH tunnel. In the example below, **SSH\_server\_IP** represents the IP address of the SSH server in your DMZ.

```
sudo ssh azureuser@<SSH_server_IP> -i ~/.ssh/id_ssh_key -N -f -L 443:controller.
<your_AAPonAzure_instance>.az.ansiblecloud.com:443
```

The **-L** flag makes your local system route traffic for the automation controller URL over port 443 (HTTPS).



#### NOTE

You must use port 443 on both sides of the routing path. Using a different port on the local machine causes some Ansible Automation Platform features to not function properly.

When the SSH tunnel is established and your Azure routing is configured, you can access the automation controller URL from your local browser at `https://controller.<your_AAPonAzure_instance>.az.ansiblecloud.com`.

