Red Hat

Plan

Ansible Automation Platform 2.6

May 12, 2026

Contents

1 Plan

Choose a deployment method and topology

Red Hat tests Ansible Automation Platform with a defined set of topologies to give you opinionated deployment options. Deploy all components of Ansible Automation Platform so that all features and capabilities are available for use without the need to take further action.

Red Hat tests the installation of Ansible Automation Platform using a defined set of infrastructure topologies or reference architectures. Enterprise organizations can use one of the enterprise topologies for production deployments. This ensures the highest level of uptime, performance, and continued scalability. Organizations or deployments that are resource constrained can use a growth topology.

You can install Ansible Automation Platform on different infrastructure topologies and environment configurations. However, Red Hat does not fully test topologies outside of published reference architectures. Red Hat recommends using a tested topology for all new deployments and provides commercially reasonable support for deployments that meet minimum requirements.

Installation and deployment models

Ansible Automation Platform offers many installation and deployment options based on your infrastructure and organizational needs. Each installation type reference includes supported infrastructure types and links to tested topologies.

NOTE:
The Ansible Automation Platform RPM installer was deprecated in 2.5 and will be removed in Ansible Automation Platform 2.7. The RPM installer will be supported for RHEL 9 during the lifecycle of Ansible Automation Platform 2.6 to support migrations to existing supported topologies. For more information on upgrade and migration paths, see the Supported upgrade and migration scenarios.

Ansible Automation Platform installation and deployment models

Mode	Infrastructure	Description	Tested topologies
Containers	Virtual machines and bare metal	The containerized installer deploys Ansible Automation Platform on Red Hat Enterprise Linux by using	• See "Container growth

Mode	Infrastructure	Description	Tested topologies
		Podman which runs the platform in containers on host machines. Customers manage the product and infrastructure lifecycle.	topology" on page 15 • See "Container enterprise topology" on page 116
Operator	Red Hat OpenShift	The Operator uses Red Hat OpenShift Operators to deploy Ansible Automation Platform within Red Hat OpenShift. Customers manage the product and infrastructure lifecycle.	• See "Operator growth topology" on page 127 • See "Operator enterprise topology" on page 134
RPM	Virtual machines and bare metal	The RPM installer deploys Ansible Automation Platform on Red Hat Enterprise Linux by using RPMs to install the platform on host machines. Customers manage the product and infrastructure lifecycle.	• See "RPM growth topology" on page 141 • See "RPM enterprise topology" on page 147

Container growth topology

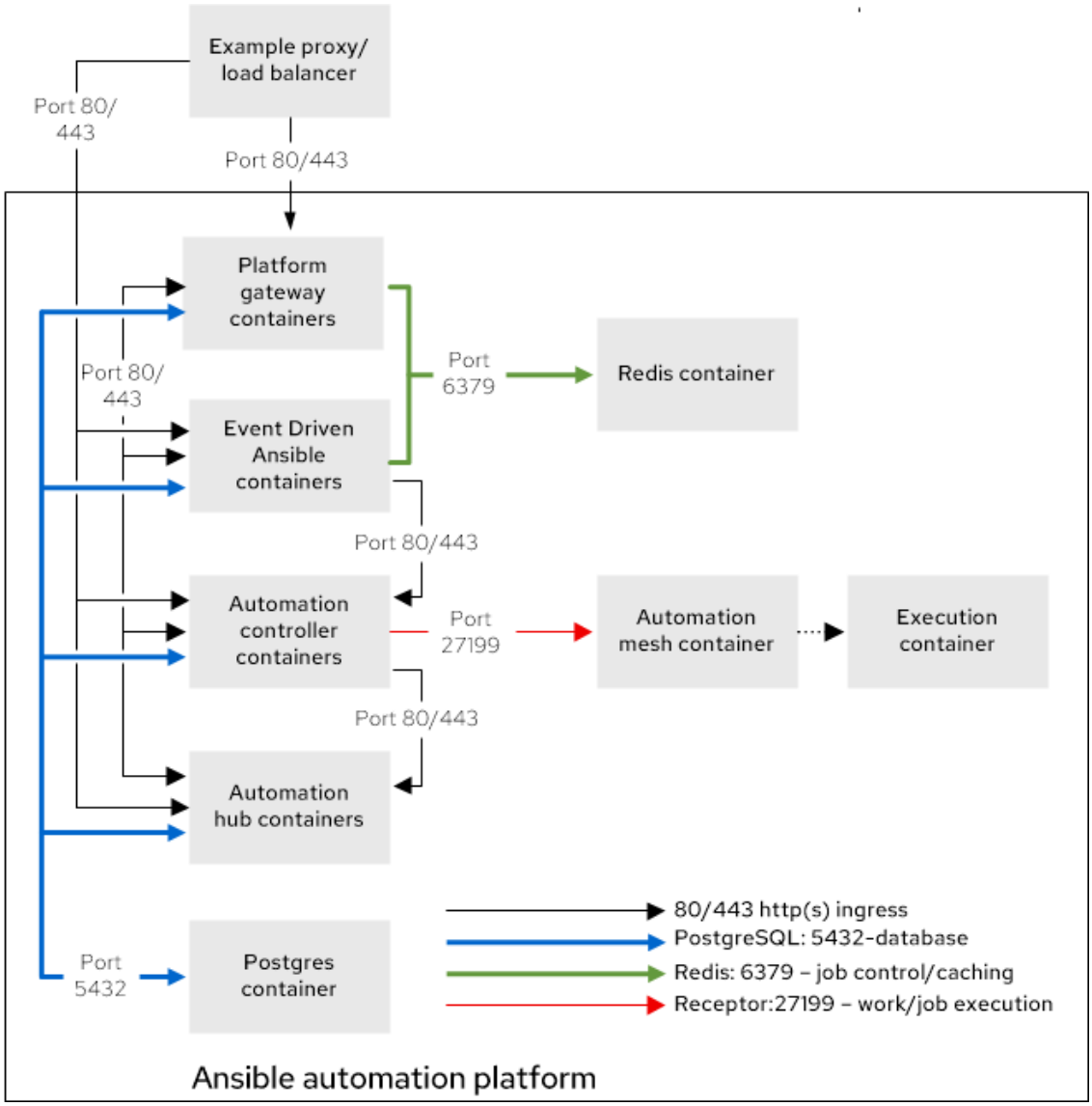
The container-based growth topology provides a smaller footprint deployment without redundancy for organizations getting started with Ansible Automation Platform.

Included are the tested infrastructure topology, system requirements, network port configurations, and an example inventory file for installation.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



Red Hat tests a single VM with these requirements:

Virtual machine requirements

Requirement	Minimum requirement
RAM	• 16 GB • 32 GB required for growth topology bundled installations with `hub_seed_collections=true`.

Requirement	Minimum requirement
	Seeding the collections can take 45 or more minutes.
CPUs	4
Local disk	• Total available disk space: 60 GB • Installation directory: 15 GB (if on a dedicated partition) • `/var/tmp` for online installations: 1 GB • `/var/tmp` for offline or bundled installations: 3 GB • Temporary directory (defaults to `/tmp`) for offline or bundled installations: 10GB
Disk IOPS	3000

Infrastructure topology components

Purpose	Example group names
All Ansible Automation Platform components	• `automationgateway` • `automationcontroller` • `automationhub` • `automationeda` • `database`

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

System configuration

Type	Description	Notes
Subscription	• Valid Red Hat Ansible Automation Platform subscription • Valid Red Hat Enterprise Linux subscription (to consume the BaseOS and AppStream repositories)	
Operating system	• Red Hat Enterprise Linux 9.4 or later minor versions of Red Hat Enterprise Linux 9. • Red Hat Enterprise Linux 10 or later minor versions of Red Hat Enterprise Linux 10.	
CPU architecture	x86_64, AArch64, s390x (IBM Z), ppc64le (IBM Power)	
`ansible-core`	• RHEL 9: installation program uses `ansible-core` 2.14, Ansible Automation Platform operation uses `ansible-core 2.16` • RHEL 10: installation program uses `ansible-core` 2.16, Ansible Automation Platform operation uses `ansible-core` 2.16	• The installation program uses the `ansible-core` package from the RHEL AppStream repository. • Ansible Automation Platform bundles `ansible-core` 2.16 for operation, so you do not need to install it manually.

Type	Description	Notes
Browser	A currently supported version of Mozilla Firefox or Google Chrome.	
Database	• For Ansible Automation Platform managed databases: PostgreSQL 15. • For customer provided (external) databases: PostgreSQL 15, 16, or 17.	• External (customer supported) databases require International Components for Unicode (ICU) support. • External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.
IP version	IPv4, IPv6 (single-stack and dual-stack)	

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination	Description
80/443	TCP	HTTP/ HTTPS	Event-Driven Ansible	Automation hub	Pull container decision environments

Port number	Protocol	Service	Source	Destination	Description
80/443	TCP	HTTP/ HTTPS	Event-Driven Ansible	Automation controller	Launch automation controller jobs
80/443	TCP	HTTP/ HTTPS	Automation controller	Automation hub	Pull collections and execution environment images
80/443	TCP	HTTP/ HTTPS	Platform gateway	Automation controller	Platform gateway to automation controller communication
80/443	TCP	HTTP/ HTTPS	Platform gateway	Automation hub	Platform gateway to automation hub communication
80/443	TCP	HTTP/ HTTPS	Platform gateway	Event-Driven Ansible	Platform gateway to Event-Driven Ansible communication
5432	TCP	PostgreSQL	Event-Driven Ansible	Database	Event-Driven Ansible database access
5432	TCP	PostgreSQL	Platform gateway	Database	Platform gateway database access
5432	TCP	PostgreSQL	Automation hub	Database	Automation hub

Port number	Protocol	Service	Source	Destination	Description
					database access
5432	TCP	PostgreSQL	Automation controller	Database	Automation controller database access
6379	TCP	Redis	Event-Driven Ansible	Redis container	Job launching and data storage for Event-Driven Ansible
6379	TCP	Redis	Platform gateway	Redis container	Data storage and retrieval for platform gateway services
8443	TCP	HTTPS	Platform gateway	Platform gateway	Internal gateway NGINX communication
27199	TCP	Receptor	Automation controller	Execution container	Mesh nodes connect directly to controllers. Allows two-way communication for job distribution.
8080/8443	TCP	HTTP/HTTPS	Platform gateway	Automation controller	Automation controller NGINX ports. You can configure these ports with the following

Port number	Protocol	Service	Source	Destination	Description
					inventory variables: `controller_nginx_http_port`, `controller_nginx_https_port`.
8081/8444	TCP	HTTP/ HTTPS	Platform gateway	Automation hub	Automation hub NGINX ports. You can configure these ports with the following inventory variables: `hub_nginx_http_port`, `hub_nginx_https_port`.
8082/8445	TCP	HTTP/ HTTPS	Platform gateway	Event-Driven Ansible	Event-Driven Ansible NGINX ports. You can configure these ports with the following inventory variables: `eda_nginx_http_port`, `eda_nginx_https_port`.
8083/8446	TCP	HTTP/ HTTPS	Platform gateway	Platform gateway	Platform gateway NGINX

Port number	Protocol	Service	Source	Destination	Description
					ports. You can configure these ports with the following inventory variables: `gateway_nginx_http_port`, `gateway_nginx_https_port`.

NOTE:
If you change any port values by using inventory variables, refer to Inventory file variables to review all default port values and ensure there are no port conflicts.

Example inventory file

Use the example inventory file to perform an installation:

```
# This is the Ansible Automation Platform installer inventory file intended for the
container growth deployment topology.

# This inventory file expects to be run from the host where Ansible Automation
Platform will be installed.

# Consult the Ansible Automation Platform product documentation about this
topology's tested hardware configuration.

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
plan-ref_installation_deployment_models

#

# Consult the docs if you are unsure what to add

# For all optional variables consult the included README.md

# or the Ansible Automation Platform documentation:

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-con_aap_containerized_installation_intro


# This section is for your platform gateway hosts

# -----------------------------------------------------
```

```
[automationgateway]

aap.example.org


# This section is for your automation controller hosts

# -----------------------------------------------------

[automationcontroller]

aap.example.org


# This section is for your automation hub hosts

# -----------------------------------------------------

[automationhub]

aap.example.org


# This section is for your Event-Driven Ansible controller hosts

# -----------------------------------------------------

[automationeda]

aap.example.org


# This section is for the Ansible Automation Platform database

# -----------------------------------------------------

[database]

aap.example.org


[all:vars]

# Ansible

ansible_connection=local


# Common variables

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_general_inventory_variables

# -----------------------------------------------------

postgresql_admin_username=postgres

postgresql_admin_password=<set your own>


registry_username=<your RHN username>

registry_password=<your RHN password>
```

```
redis_mode=standalone


# Platform gateway

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_gateway_variables

# ------------------------------------------------------

gateway_admin_password=<set your own>

gateway_pg_host=aap.example.org

gateway_pg_password=<set your own>


# Automation controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_controller_variables

# ------------------------------------------------------

controller_admin_password=<set your own>

controller_pg_host=aap.example.org

controller_pg_password=<set your own>

controller_percent_memory_capacity=0.5


# Automation hub

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_hub_variables

# ------------------------------------------------------

hub_admin_password=<set your own>

hub_pg_host=aap.example.org

hub_pg_password=<set your own>

hub_seed_collections=false


# Event-Driven Ansible controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_eda_controller_variables

# ------------------------------------------------------

eda_admin_password=<set your own>

eda_pg_host=aap.example.org

eda_pg_password=<set your own>
```

SSH keys are only required when installing on remote hosts. If doing a self contained local VM based installation, you can use `ansible_connection=local`.

Container enterprise topology

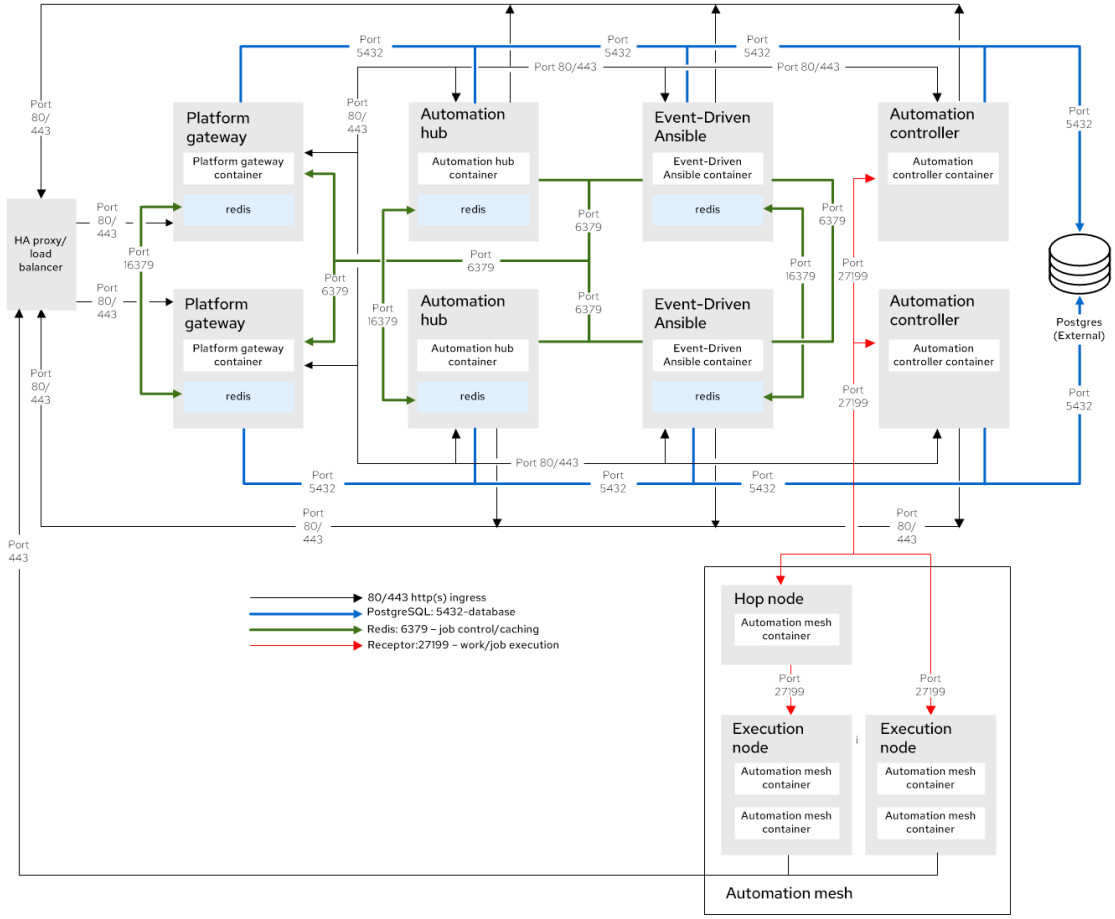
The container-based enterprise topology provides redundancy and higher compute for large volumes of automation.

Included are the tested infrastructure topology, system requirements, network port configurations, and an example inventory file for installation.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



Red Hat tests each VM with these requirements:

Virtual machine requirements

Requirement	Minimum requirement
RAM	16 GB

Requirement	Minimum requirement
CPUs	4
Local disk	• Total available disk space: 60 GB • Installation directory: 15 GB (if on a dedicated partition) • `/var/tmp` for online installations: 1 GB • `/var/tmp` for offline or bundled installations: 3 GB • Temporary directory (defaults to `/tmp`) for offline or bundled installations: 10GB
Disk IOPS	3000

Infrastructure topology components

VM count	Purpose	Example VM group names
2	Platform gateway with colocated Redis	`automationgateway`
2	Automation controller	`automationcontroller`
2	Private automation hub with colocated Redis	`automationhub`
2	Event-Driven Ansible with colocated Redis	`automationeda`
1	Automation mesh hop node	`execution_nodes`
2	Automation mesh execution node	`execution_nodes`
1	Externally managed database service	N/A
1	HAProxy load balancer in front of platform gateway (externally managed)	N/A

NOTE:

- 6 VMs are required for a Redis high availability (HA) compatible deployment. When installing Ansible Automation Platform with the containerized installer, Redis can be colocated on any Ansible Automation Platform component VMs of your choice except for execution nodes or the PostgreSQL database. They might also be assigned VMs specifically for Redis use.
- External Redis is not supported for containerized Ansible Automation Platform.

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

System configuration

Type	Description	Notes
Subscription	• Valid Red Hat Ansible Automation Platform subscription • Valid Red Hat Enterprise Linux subscription (to consume the BaseOS and AppStream repositories)	
Operating system	• Red Hat Enterprise Linux 9.4 or later minor versions of Red Hat Enterprise Linux 9. • Red Hat Enterprise Linux 10 or later minor versions of Red Hat Enterprise Linux 10.	
CPU architecture	x86_64, AArch64, s390x (IBM Z), ppc64le (IBM Power)	

Type	Description	Notes
`ansible-core`	• RHEL 9: installation program uses `ansible-core` 2.14, Ansible Automation Platform operation uses `ansible-core` 2.16. • RHEL 10: installation program uses `ansible-core` 2.16, Ansible Automation Platform operation uses `ansible-core` 2.16.	• The installation program uses the `ansible-core` package from the RHEL AppStream repository. • Ansible Automation Platform bundles `ansible-core` 2.16 for operation, so you do not need to install it manually.
Browser	A currently supported version of Mozilla Firefox or Google Chrome.	
Database	• For Ansible Automation Platform managed databases: PostgreSQL 15. • For customer provided (external) databases: PostgreSQL 15, 16, or 17.	• External (customer supported) databases require International Components for Unicode (ICU) support. • External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.
IP version	IPv4, IPv6 (single-stack and dual-stack)	

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination	Description
80/443	TCP	HTTP/ HTTPS	Event-Driven Ansible	Automation hub	Pull container decision environments
80/443	TCP	HTTP/ HTTPS	Event-Driven Ansible	Automation controller	Launch automation controller jobs
80/443	TCP	HTTP/ HTTPS	Automation controller	Automation hub	Pull collections and execution environment images
80/443	TCP	HTTP/ HTTPS	HAProxy load balancer	Platform gateway	External load balancer access
80/443	TCP	HTTP/ HTTPS	Platform gateway	Automation controller	Platform gateway to automation controller communication
80/443	TCP	HTTP/ HTTPS	Platform gateway	Automation hub	Platform gateway to automation hub communication

Port number	Protocol	Service	Source	Destination	Description
80/443	TCP	HTTP/ HTTPS	Platform gateway	Event-Driven Ansible	Platform gateway to Event-Driven Ansible communication
5432	TCP	PostgreSQL	Event-Driven Ansible	External database	Event-Driven Ansible database access
5432	TCP	PostgreSQL	Platform gateway	External database	Platform gateway database access
5432	TCP	PostgreSQL	Automation hub	External database	Automation hub database access
5432	TCP	PostgreSQL	Automation controller	External database	Automation controller database access
6379	TCP	Redis	Event-Driven Ansible	Redis node	Job launching and data storage for Event-Driven Ansible
6379	TCP	Redis	Platform gateway	Redis node	Data storage and retrieval for platform gateway services
16379	TCP	Redis	Redis node	Redis node	Redis cluster bus communication

Port number	Protocol	Service	Source	Destination	Description
27199	TCP	Receptor	Automation controller	Hop node and execution node	Mesh nodes connect directly to controllers. Allows two-way communication for job distribution.
27199	TCP	Receptor	Hop node	Execution node	Mesh nodes connect through hop nodes. Allows two-way communication in either direction.
8080/8443	TCP	HTTP/ HTTPS	Platform gateway	Automation controller	Automation controller NGINX ports. You can configure these ports with the following inventory variables: `controller_nginx_http_port`, `controller_nginx_https_port`.
8081/8444	TCP	HTTP/ HTTPS	Platform gateway	Automation hub	Automation hub NGINX ports. You can configure these ports with the

Port number	Protocol	Service	Source	Destination	Description
					following inventory variables: `hub_nginx_http_port`, `hub_nginx_https_port`.
8082/8445	TCP	HTTP/ HTTPS	Platform gateway	Event-Driven Ansible	Event-Driven Ansible NGINX ports. You can configure these ports with the following inventory variables: `eda_nginx_http_port`, `eda_nginx_https_port`.
8083/8446	TCP	HTTP/ HTTPS	Platform gateway	Platform gateway	Platform gateway NGINX ports. You can configure these ports with the following inventory variables: `gateway_nginx_http_port`, `gateway_nginx_https_port`.

NOTE:
If you change any port values by using inventory variables, refer to Inventory file variables to review all default port values and ensure there are no port conflicts.

Example inventory file

Use the example inventory file to perform an installation:

```
# This is the Ansible Automation Platform enterprise installer inventory file
# Consult the docs if you are unsure what to add
# For all optional variables consult the included README.md
# or the Red Hat documentation:
# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-con_aap_containerized_installation_intro


# This section is for your platform gateway hosts
# -----------------------------------------------------
[automationgateway]
gateway1.example.org
gateway2.example.org


# This section is for your automation controller hosts
# -----------------------------------------------------
[automationcontroller]
controller1.example.org
controller2.example.org


# This section is for your Ansible Automation Platform execution hosts
# -----------------------------------------------------
[execution_nodes]
hop1.example.org receptor_type='hop'
exec1.example.org
exec2.example.org


# This section is for your automation hub hosts
# -----------------------------------------------------
```

```
[automationhub]

hub1.example.org

hub2.example.org


# This section is for your Event-Driven Ansible controller hosts

# -----------------------------------------------------

[automationeda]

eda1.example.org

eda2.example.org


[redis]

gateway1.example.org

gateway2.example.org

hub1.example.org

hub2.example.org

eda1.example.org

eda2.example.org


[all:vars]


# Common variables

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_general_variables

# -----------------------------------------------------

postgresql_admin_username=<set your own>

postgresql_admin_password=<set your own>

registry_username=<your RHN username>

registry_password=<your RHN password>


# Platform gateway

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_gateway_variables

# -----------------------------------------------------

gateway_admin_password=<set your own>

gateway_pg_host=externaldb.example.org

gateway_pg_database=<set your own>

gateway_pg_username=<set your own>
```

```
gateway_pg_password=<set your own>


# Automation controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_controller_variables

# ------------------------------------------------------

controller_admin_password=<set your own>

controller_pg_host=externaldb.example.org

controller_pg_database=<set your own>

controller_pg_username=<set your own>

controller_pg_password=<set your own>


# Automation hub

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_hub_variables

# ------------------------------------------------------

hub_admin_password=<set your own>

hub_pg_host=externaldb.example.org

hub_pg_database=<set your own>

hub_pg_username=<set your own>

hub_pg_password=<set your own>


# Event-Driven Ansible controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
install-ref_eda_controller_variables

# ------------------------------------------------------

eda_admin_password=<set your own>

eda_pg_host=externaldb.example.org

eda_pg_database=<set your own>

eda_pg_username=<set your own>

eda_pg_password=<set your own>
```

Operator growth topology

The Operator-based growth topology provides a smaller footprint deployment without redundancy for organizations getting started with Ansible Automation Platform on Red Hat OpenShift Container Platform.

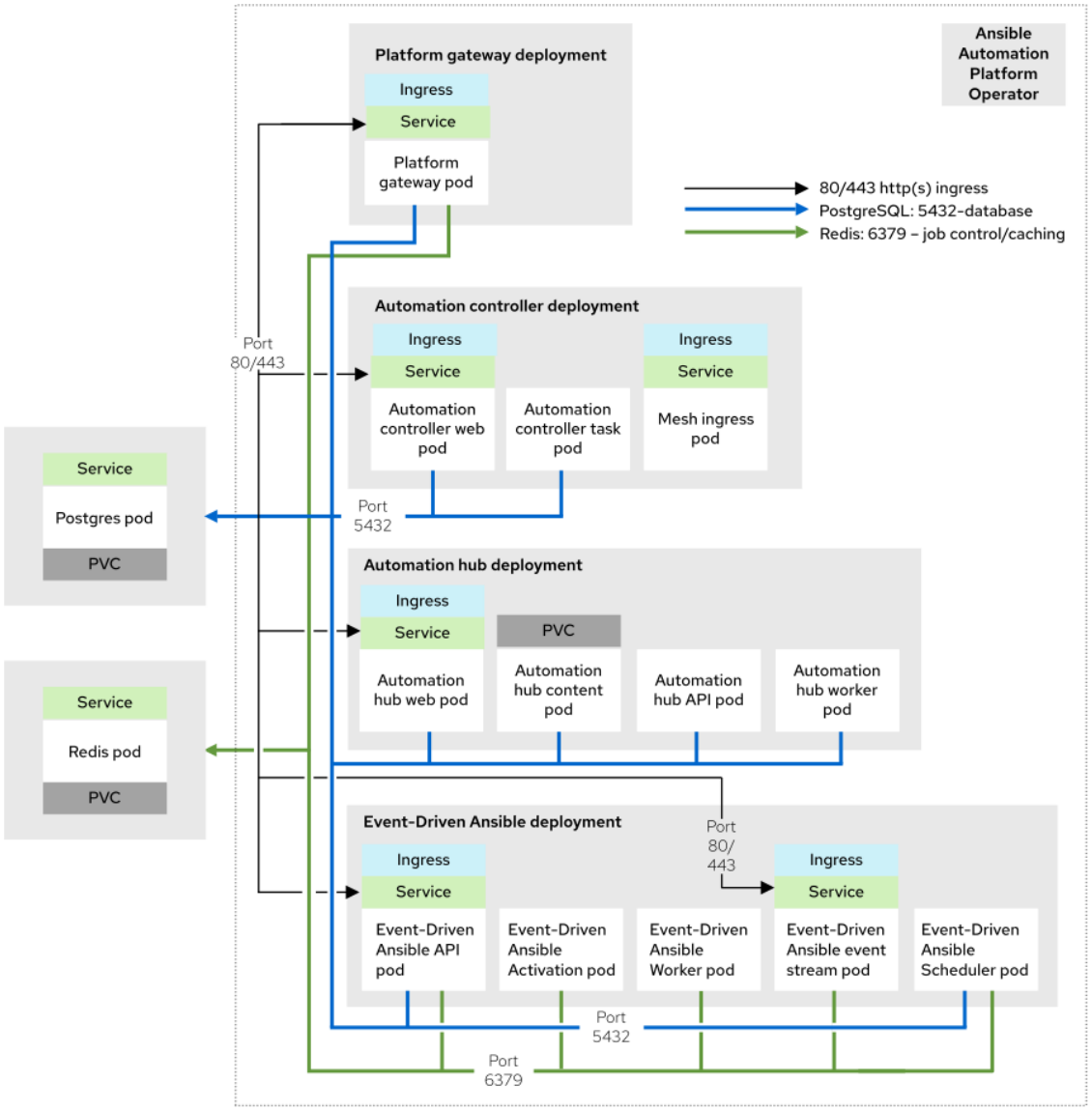
Included are the tested infrastructure topology, system requirements, network port configurations, and an example custom resource file for installation.

IMPORTANT:
You can only install a single instance of the Ansible Automation Platform Operator into a single namespace. Installing multiple instances in the same namespace can lead to improper operation for both Operator instances.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



IMPORTANT:
While Redis and PostgreSQL can be installed as part of the operator-based installation process, the topology diagram represents a Red Hat supported topology where both Redis and PostgreSQL are external to Ansible Automation Platform.

Red Hat tests a Single Node OpenShift (SNO) cluster with these requirements: 32 GB RAM, 16 CPUs, 128 GB local disk, and 3000 IOPS.

Infrastructure topology components

Count	Component
1	Automation controller web pod
1	Automation controller task pod

Count	Component
1	Automation hub web pod
1	Automation hub API pod
2	Automation hub content pod
2	Automation hub worker pod
1	Automation hub Redis pod
1	Event-Driven Ansible API pod
1	Event-Driven Ansible activation worker pod
1	Event-Driven Ansible default worker pod
1	Event-Driven Ansible event stream pod
1	Event-Driven Ansible scheduler pod
1	Platform gateway pod
1	Database pod
1	Redis pod

NOTE:
You can deploy multiple isolated instances of Ansible Automation Platform into the same Red Hat OpenShift Container Platform cluster. To do this, use a namespace-scoped deployment model (isolated within a namespace).

This approach allows you to use the same cluster for several deployments.

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

Tested system configurations

Type	Description	Notes
Subscription	Valid Red Hat Ansible Automation Platform subscription	

Type	Description	Notes
Red Hat OpenShift	• `Version`: 4.14 • `num_of_control_nodes`: 1 • `num_of_worker_nodes`: 1	
Ansible-core	Ansible-core version 2.16 or later	
Browser	A currently supported version of Mozilla Firefox or Google Chrome.	
Database	• For Ansible Automation Platform managed databases: PostgreSQL 15 • For customer provided (external) databases: PostgreSQL 15, 16, or 17.	• External (customer supported) databases require International Components for Unicode (ICU) support. • External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality depends on utilities provided with PostgreSQL 15.

Type	Description	Notes
		WARNING: Operator-deployed database connection limits The operator-deployed PostgreSQL database has a default `max_connections` limit of 100 and a maximum database size of 100 GB. Do not use the operator-deployed database if any of the following conditions apply: • You are running more than 1 replica of any component (platform gateway, automation controller, automation hub, or Event-dDriven Ansible). • The automation controller capacity exceeds 100 concurrent jobs. • Total database connections from all components exceed 100. • Estimated 30-day database storage exceeds 100 GB.

Type	Description	Notes
		If your deployment exceeds any of these limits, use an external database instead of the operator-deployed database. Database storage consumption depends on your workload, including job frequency, playbook task count, output verbosity, and the number of managed hosts per job. Higher verbosity levels can increase storage consumption by 3-5x.
IP version	IPv4, IPv6 (single-stack and dual-stack)	

Example custom resource file

Use this example custom resource (CR) to add your Ansible Automation Platform instance to your project:

```
apiVersion: aap.ansible.com/v1alpha1

kind: AnsibleAutomationPlatform

metadata:

  name: <aap instance name>

spec:

  eda:

    automation_server_ssl_verify: 'no'

  hub:

    storage_type: 's3'

    object_storage_s3_secret: '<name of the Secret resource holding s3
configuration>'
```

Nonfunctional requirements

Ansible Automation Platform's performance characteristics and capacity depend on its resource allocation and configuration. With OpenShift, each Ansible Automation Platform component deploys as a pod. You can specify resource requests and limits for each pod.

Use the Ansible Automation Platform Custom Resource (CR) to configure resource allocation for OpenShift installations. Each configurable item has default settings. These settings are the minimum requirements for an installation, but might not meet your production workload needs.

By default, each component's deployments use minimum resource requests but no resource limits. OpenShift only schedules pods with available resource requests, but the pods can consume unlimited RAM or CPU as long as the OpenShift worker node itself is not under node pressure.

In the Operator growth topology, Ansible Automation Platform runs on a Single Node OpenShift (SNO) with 32 GB RAM, 16 CPUs, 128 GB local disk, and 3000 IOPS. This is not a shared environment, so Ansible Automation Platform pods have full access to all of the compute resources of the OpenShift SNO. In this scenario, the capacity calculation for automation controller task pods comes from the underlying OpenShift Container Platform node that runs the pod. It does not have access to the entire node. This capacity calculation influences how many concurrent jobs automation controller can run.

OpenShift manages storage distinctly from VMs. This impacts how automation hub stores its artifacts. In the Operator growth topology, the topology uses S3 storage because automation hub requires a `ReadWriteMany` type storage, which is not a default storage type in OpenShift.

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination
80/443	HTTP/HTTPS	Receptor	Execution node	OpenShift Container Platform ingress
80/443	HTTP/HTTPS	Receptor	Hop node	OpenShift Container Platform ingress

Port number	Protocol	Service	Source	Destination
80/443	HTTP/HTTPS	Platform	Customer clients	OpenShift Container Platform ingress
27199	TCP	Receptor	OpenShift Container Platform cluster	Execution node
27199	TCP	Receptor	OpenShift Container Platform cluster	Hop node

Operator enterprise topology

The Operator-based enterprise topology provides redundancy and higher compute for large volumes of automation on Red Hat OpenShift Container Platform.

The Ansible Automation Platform Service on AWS is an example of an OpenShift Operator based enterprise topology.

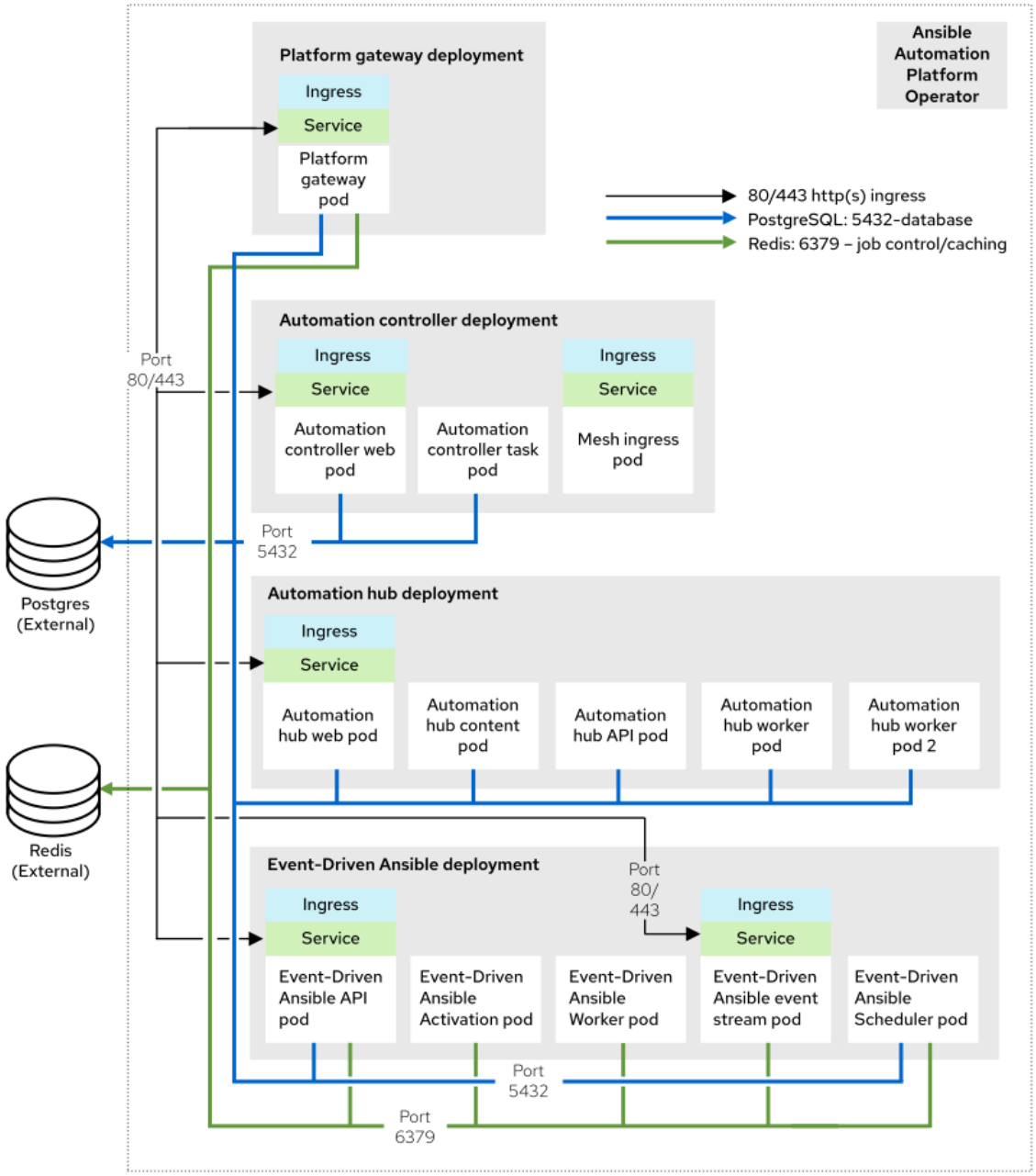
Included are the tested infrastructure topology, system requirements, network port configurations, and an example custom resource file for installation.

IMPORTANT:
You can only install a single instance of the Ansible Automation Platform Operator into a single namespace. Installing multiple instances in the same namespace can lead to improper operation for both Operator instances.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



IMPORTANT:
While Redis and PostgreSQL can be installed as part of the operator-based installation process, the topology diagram represents a Red Hat supported topology where both Redis and PostgreSQL are external to Ansible Automation Platform.

This infrastructure topology describes an OpenShift Cluster with 3 primary nodes and 2 worker nodes.

Red Hat tests each OpenShift Worker node with these requirements: 16 GB RAM, 4 CPUs, 128 GB local disk, and 3000 IOPS.

Infrastructure topology components

Count	Component
1	Automation controller web pod
1	Automation controller task pod
1	Automation hub web pod
1	Automation hub API pod
2	Automation hub content pod
2	Automation hub worker pod
1	Automation hub Redis pod
1	Event-Driven Ansible API pod
2	Event-Driven Ansible activation worker pod
2	Event-Driven Ansible default worker pod
2	Event-Driven Ansible event stream pod
1	Event-Driven Ansible scheduler pod
1	Platform gateway pod
2	Mesh ingress pod
N/A	Externally managed database service
N/A	Externally managed Redis
N/A	Externally managed object storage service (for automation hub)

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

Tested system configurations

Type	Description
Subscription	Valid Red Hat Ansible Automation Platform subscription
Red Hat OpenShift	• Red Hat OpenShift on AWS Hosted Control Planes 4.15.16 ◦ 2 worker nodes in different availability zones (AZs) at t3.xlarge
Ansible-core	Ansible-core version 2.16 or later
Browser	A currently supported version of Mozilla Firefox or Google Chrome.
AWS RDS PostgreSQL service	• engine: "postgres" • engine_version: 15" • parameter_group_name: "default.postgres15" • allocated_storage: 20 • max_allocated_storage: 1000 • storage_type: "gp2" • storage_encrypted: true • instance_class: "db.t4g.small" • multi_az: true • backup_retention_period: 5 • database: must have International Components for Unicode (ICU) support

Type	Description
	NOTE: Minimum external database requirements The external database must meet these minimum requirements: • 4 vCPUs • 16 GB RAM • max_connections: 1024 (minimum). You might need more connections when scaling replicas. • 200 GB storage on a volume capable of at least 3000 IOPS. Database storage consumption depends on your workload, including job frequency, playbook task count, output verbosity, and the number of managed hosts per job. Start with a 200 GB baseline and monitor actual usage after deployment. Configure automated cleanup jobs to prevent unbounded database growth. These requirements ensure adequate database performance for the enterprise topology workload profile.
AWS Memcached Service	• engine: "redis" • engine_version: "6.2" • auto_minor_version_upgrade: "false" • node_type: "cache.t3.micro" • parameter_group_name: "default.redis6.x.cluster.on" • transit_encryption_enabled: "true" • num_node_groups: 2 • replicas_per_node_group: 1 • automatic_failover_enabled: true

Type	Description
s3 storage	HTTPS only accessible through AWS Role assigned to automation hub SA at runtime by using AWS Pod Identity
IP version	IPv4, IPv6 (single-stack and dual-stack)

Example custom resource file

For example CR files, see the ocp-b.env-a directory in the `test-topologies` GitHub repository.

Nonfunctional requirements

Ansible Automation Platform's performance characteristics and capacity depend on its resource allocation and configuration. With OpenShift, each Ansible Automation Platform component deploys as a pod. You can specify resource requests and limits for each pod.

Use the Ansible Automation Platform custom resource to configure resource allocation for OpenShift installations. Each configurable item has default settings. These settings are the exact configuration used in this reference deployment architecture. This configuration assumes deployment and management by an Enterprise IT organization for production purposes.

By default, each component's deployments use minimum resource requests but no resource limits. OpenShift only schedules pods with available resource requests. However, pods can consume unlimited RAM or CPU as long as the OpenShift worker node is not under node pressure.

In the Operator enterprise topology, Ansible Automation Platform runs on a Red Hat OpenShift on AWS (ROSA) Hosted Control Plane (HCP) cluster. The cluster has 2 t3.xlarge worker nodes spread across 2 AWS availability zones within a single region. This is not a shared environment so Ansible Automation Platform pods have full access to all compute resources of the ROSA HCP cluster.

The capacity calculation for automation controller task pods comes from the underlying HCP worker node running the pod. It does not have access to the CPU or memory resources of the entire node. This capacity calculation influences how many concurrent jobs automation controller can run.

OpenShift manages storage distinctly from VMs. This impacts how automation hub stores its artifacts. In the Operator enterprise topology, automation hub uses S3 storage. automation hub requires `ReadWriteMany` type storage, which is not a default storage type in OpenShift.

This topology specifies externally provided Redis, PostgreSQL, and object storage for automation hub. This provides additional scalability and reliability features for the Ansible Automation

Platform deployment. These features include specialized backup, restore, and replication services, as well as scalable storage.

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination
80/443	HTTP/HTTPS	Object storage	OpenShift Container Platform cluster	External object storage service
80/443	HTTP/HTTPS	Receptor	Execution node	OpenShift Container Platform ingress
80/443	HTTP/HTTPS	Receptor	Hop node	OpenShift Container Platform ingress
5432	TCP	PostgreSQL	OpenShift Container Platform cluster	External database service
6379	TCP	Redis	OpenShift Container Platform cluster	External Redis service
27199	TCP	Receptor	OpenShift Container Platform cluster	Execution node
27199	TCP	Receptor	OpenShift Container Platform cluster	Hop node

RPM growth topology

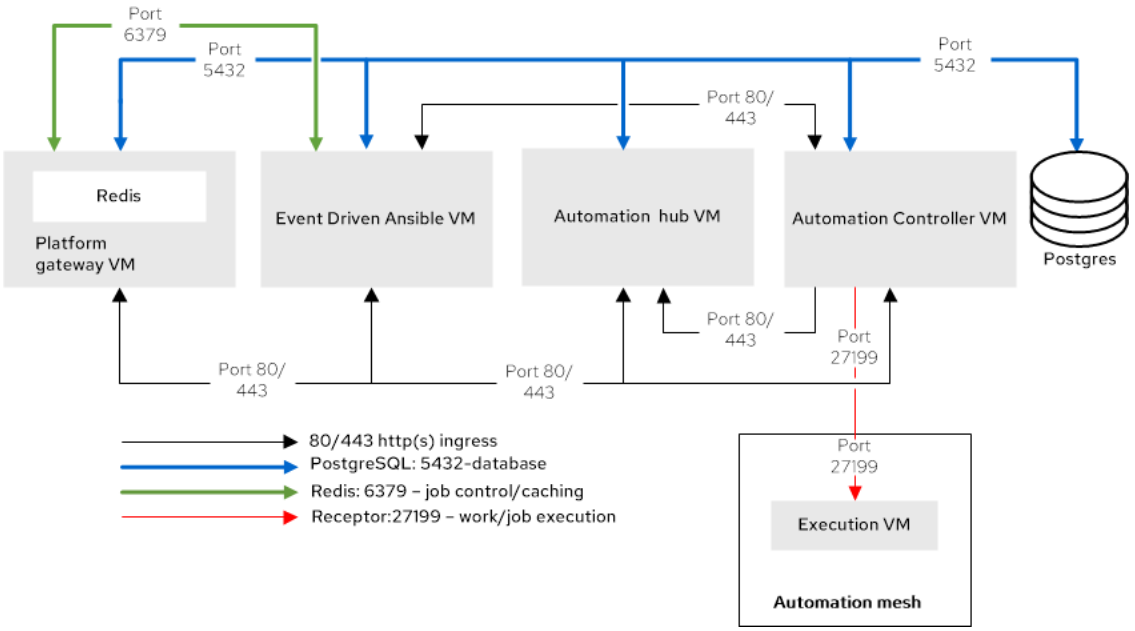
The RPM-based growth topology provides a smaller footprint deployment without redundancy for organizations getting started with Ansible Automation Platform.

Included are the tested infrastructure topology, system requirements, network port configurations, and an example inventory file for installation.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



Red Hat tests each VM with these requirements:

Virtual machine requirements

Requirement	Minimum requirement
RAM	16 GB
CPUs	4
Local disk	60 GB
Disk IOPS	3000

Infrastructure topology components

VM count	Purpose	Example VM group names
1	Platform gateway with colocated Redis	`automationgateway`
1	Automation controller	`automationcontroller`
1	Private automation hub	`automationhub`
1	Event-Driven Ansible	`automationedacontroller`
1	Automation mesh execution node	`execution_nodes`
1	Ansible Automation Platform managed database	`database`

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

Tested system configurations

Type	Description	
Subscription	Valid Red Hat Ansible Automation Platform subscription	
Operating system	Red Hat Enterprise Linux 9.4 or later minor versions of Red Hat Enterprise Linux 9.	
CPU architecture	x86_64, AArch64, s390x (IBM Z), ppc64le (IBM Power)	
`ansible-core`	`ansible-core` version 2.16 or later	Ansible Automation Platform uses the system-wide ansible-core package to install the platform, but uses ansible-core 2.16 for both its control plane and

Type	Description	
		built-in execution environments.
Browser	A currently supported version of Mozilla Firefox or Google Chrome	
Database	• For Ansible Automation Platform managed databases: PostgreSQL 15 • For customer provided (external) databases: PostgreSQL 15, 16, or 17.	• External (customer supported) databases require ICU support. • External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.
IP version	IPv4, IPv6 (single-stack and dual-stack)	

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination
80/443	TCP	HTTP/HTTPS	Event-Driven Ansible	Automation hub
80/443	TCP	HTTP/HTTPS	Event-Driven Ansible	Automation controller
80/443	TCP	HTTP/HTTPS	Automation controller	Automation hub

Port number	Protocol	Service	Source	Destination
80/443	TCP	HTTP/HTTPS	Platform gateway	Automation controller
80/443	TCP	HTTP/HTTPS	Platform gateway	Automation hub
80/443	TCP	HTTP/HTTPS	Platform gateway	Event-Driven Ansible
80/443	TCP	HTTP/HTTPS	Execution node	Platform gateway
5432	TCP	PostgreSQL	Event-Driven Ansible	Database
5432	TCP	PostgreSQL	Platform gateway	Database
5432	TCP	PostgreSQL	Automation hub	Database
5432	TCP	PostgreSQL	Automation controller	Database
6379	TCP	Redis	Event-Driven Ansible	Redis node
6379	TCP	Redis	Platform gateway	Redis node
8443	TCP	HTTPS	Platform gateway	Platform gateway
27199	TCP	Receptor	Automation controller	Execution node

NOTE:
If you change any port values by using inventory variables, refer to Inventory file variables to review all default port values and ensure there are no port conflicts.

Example inventory file

Use the example inventory file to perform an installation:

```
# This is the Ansible Automation Platform installer inventory file intended for the
RPM growth deployment topology.

# Consult the Ansible Automation Platform product documentation about this
topology's tested hardware configuration.

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/tested_deployment_models/rpm-topologies

#

# Consult the docs if you are unsure what to add

# For all optional variables consult the Ansible Automation Platform documentation:

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/index




# This section is for your platform gateway hosts

# -----------------------------------------------------

[automationgateway]

gateway.example.org


# This section is for your automation controller hosts

# -----------------------------------------------------

[automationcontroller]

controller.example.org


[automationcontroller:vars]

peers=execution_nodes


# This section is for your Ansible Automation Platform execution hosts

# -----------------------------------------------------

[execution_nodes]

exec.example.org


# This section is for your automation hub hosts

# -----------------------------------------------------

[automationhub]

hub.example.org


# This section is for your Event-Driven Ansible controller hosts

# -----------------------------------------------------
```

```
[automationedacontroller]

eda.example.org


# This section is for the Ansible Automation Platform database

# ------------------------------------------------------

[database]

db.example.org


[all:vars]


# Common variables

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#general-variables

# ------------------------------------------------------

registry_username=<your RHN username>

registry_password=<your RHN password>


redis_mode=standalone


# Platform gateway

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#platform-gateway-variables

# ------------------------------------------------------

automationgateway_admin_password=<set your own>

automationgateway_pg_host=db.example.org

automationgateway_pg_password=<set your own>


# Automation controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#controller-variables

# ------------------------------------------------------

admin_password=<set your own>

pg_host=db.example.org

pg_password=<set your own>


# Automation hub

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#hub-variables
```

```
# ----------------------------------------------------

automationhub_admin_password=<set your own>

automationhub_pg_host=db.example.org

automationhub_pg_password=<set your own>


# Event-Driven Ansible controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#event-driven-ansible-variables

# ----------------------------------------------------

automationedacontroller_admin_password=<set your own>

automationedacontroller_pg_host=db.example.org

automationedacontroller_pg_password=<set your own>
```

RPM enterprise topology

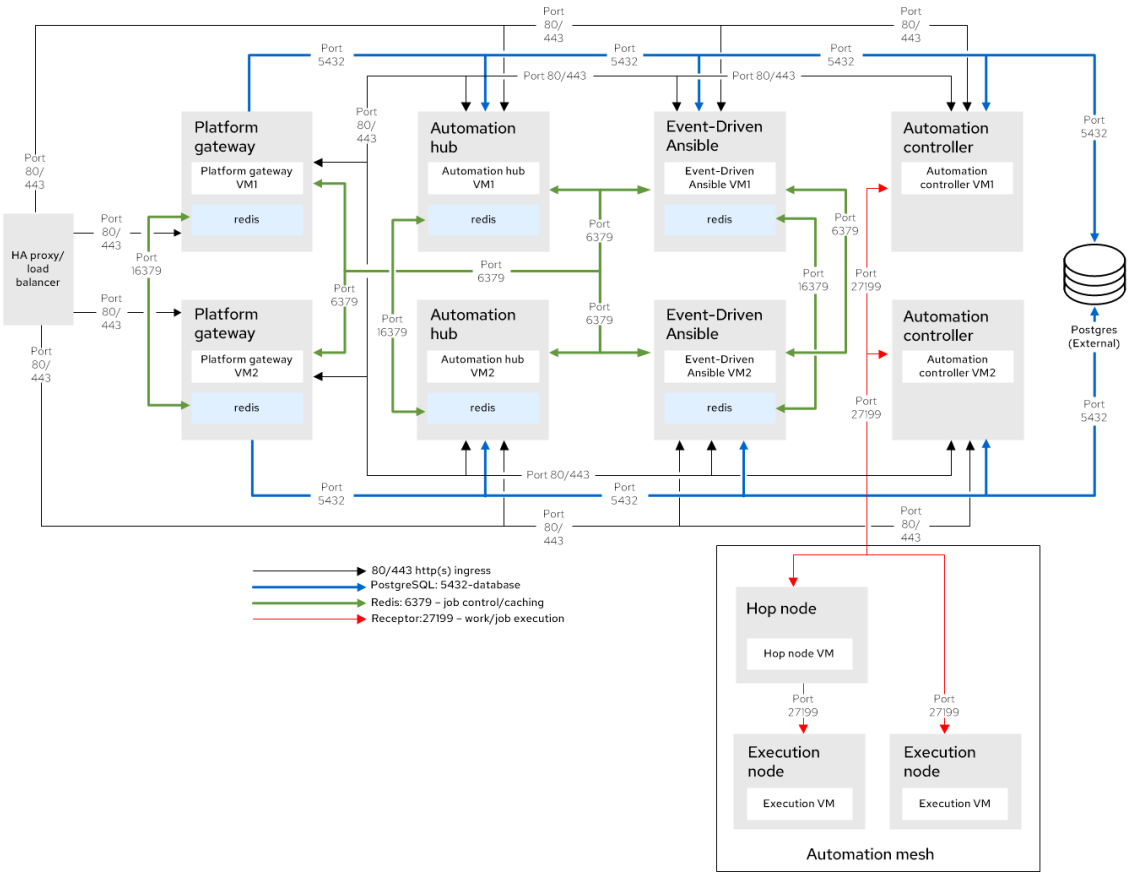
The RPM-based enterprise topology provides redundancy and higher compute for large volumes of automation.

Included are the tested infrastructure topology, system requirements, network port configurations, and an example inventory file for installation.

Infrastructure topology

The Red Hat tested infrastructure topology for this deployment model:

Figure: Infrastructure topology diagram



Red Hat tests each VM with these requirements:

Virtual machine requirements

Requirement	Minimum requirement
RAM	16 GB
CPUs	4
Local disk	60 GB
Disk IOPS	3000

Infrastructure topology components

VM count	Purpose	Example VM group names
2	Platform gateway with colocated Redis	`automationgateway`
2	Automation controller	`automationcontroller`
2	Private automation hub with colocated Redis	`automationhub`

VM count	Purpose	Example VM group names
2	Event-Driven Ansible with colocated Redis	`automationedacontroller`
1	Automation mesh hop node	`execution_nodes`
2	Automation mesh execution node	`execution_nodes`
1	Externally managed database service	N/A
1	HAProxy load balancer in front of platform gateway (externally managed)	N/A

NOTE:

- Redis high availability (HA) deployment requires 6 VMs. You can colocate Redis on each Ansible Automation Platform component VM except for automation controller, execution nodes, or the PostgreSQL database.
- RPM-based deployments of Ansible Automation Platform do not support external Redis.

Tested system configurations

Red Hat has tested these configurations to install and run Red Hat Ansible Automation Platform:

Tested system configurations

Type	Description	
Subscription	Valid Red Hat Ansible Automation Platform subscription	
Operating system	Red Hat Enterprise Linux 9.4 or later minor versions of Red Hat Enterprise Linux 9.	
CPU architecture	x86_64, AArch64, s390x (IBM Z), ppc64le (IBM Power)	

Type	Description	
`ansible-core`	`ansible-core` version 2.16 or later	Ansible Automation Platform uses the system-wide ansible-core package to install the platform, but uses ansible-core 2.16 for both its control plane and built-in execution environments.
Browser	A currently supported version of Mozilla Firefox or Google Chrome	
Database	• For Ansible Automation Platform managed databases: PostgreSQL 15 • For customer provided (external) databases: PostgreSQL 15, 16, or 17.	• External (customer supported) databases require ICU support. • External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.
IP version	IPv4, IPv6 (single-stack and dual-stack)	

Network ports

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination
80/443	TCP	HTTP/HTTPS	Event-Driven Ansible	Automation hub

Port number	Protocol	Service	Source	Destination
80/443	TCP	HTTP/HTTPS	Event-Driven Ansible	Automation controller
80/443	TCP	HTTP/HTTPS	Automation controller	Automation hub
80/443	TCP	HTTP/HTTPS	HAProxy load balancer	Platform gateway
80/443	TCP	HTTP/HTTPS	Platform gateway	Automation controller
80/443	TCP	HTTP/HTTPS	Platform gateway	Automation hub
80/443	TCP	HTTP/HTTPS	Platform gateway	Event-Driven Ansible
80/443	TCP	HTTP/HTTPS	Execution node	Platform gateway
5432	TCP	PostgreSQL	Event-Driven Ansible	External database
5432	TCP	PostgreSQL	Platform gateway	External database
5432	TCP	PostgreSQL	Automation hub	External database
5432	TCP	PostgreSQL	Automation controller	External database
6379	TCP	Redis	Event-Driven Ansible	Redis node
6379	TCP	Redis	Platform gateway	Redis node
8443	TCP	HTTPS	Platform gateway	Platform gateway
16379	TCP	Redis	Redis node	Redis node
27199	TCP	Receptor	Automation controller	Hop node and execution node
27199	TCP	Receptor	Hop node	Execution node

NOTE:
If you change any port values by using inventory variables, refer to Inventory file variables to review all default port values and ensure there are no port conflicts.

Example inventory file

Use the example inventory file to perform an installation:

```
# This is the Ansible Automation Platform enterprise installer inventory file

# Consult the docs if you are unsure what to add

# For all optional variables consult the Red Hat documentation:

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/index


# This section is for your platform gateway hosts

# -----------------------------------------------------

[automationgateway]

gateway1.example.org

gateway2.example.org


# This section is for your automation controller hosts

# -----------------------------------------------------

[automationcontroller]

controller1.example.org

controller2.example.org


[automationcontroller:vars]

peers=execution_nodes


# This section is for your Ansible Automation Platform execution hosts

# -----------------------------------------------------

[execution_nodes]

hop1.example.org node_type='hop'

exec1.example.org

exec2.example.org
```

```
# This section is for your automation hub hosts
# -----------------------------------------------------
[automationhub]
hub1.example.org
hub2.example.org


# This section is for your Event-Driven Ansible controller hosts
# -----------------------------------------------------
[automationedacontroller]
eda1.example.org
eda2.example.org


[redis]
gateway1.example.org
gateway2.example.org
hub1.example.org
hub2.example.org
eda1.example.org
eda2.example.org


[all:vars]
# Common variables
# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#general-variables
# -----------------------------------------------------
registry_username=<your RHN username>
registry_password=<your RHN password>


# Platform gateway
# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#platform-gateway-variables
# -----------------------------------------------------
automationgateway_admin_password=<set your own>
automationgateway_pg_host=<set your own>
automationgateway_pg_database=<set your own>
automationgateway_pg_username=<set your own>
automationgateway_pg_password=<set your own>
```

```
# Automation controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#controller-variables

# ------------------------------------------------------

admin_password=<set your own>

pg_host=<set your own>

pg_database=<set your own>

pg_username=<set your own>

pg_password=<set your own>


# Automation hub

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#hub-variables

# ------------------------------------------------------

automationhub_admin_password=<set your own>

automationhub_pg_host=<set your own>

automationhub_pg_database=<set your own>

automationhub_pg_username=<set your own>

automationhub_pg_password=<set your own>


# Event-Driven Ansible controller

# https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.6/
html/rpm_installation/appendix-inventory-files-vars#event-driven-ansible-variables

# ------------------------------------------------------

automationedacontroller_admin_password=<set your own>

automationedacontroller_pg_host=<set your own>

automationedacontroller_pg_database=<set your own>

automationedacontroller_pg_username=<set your own>

automationedacontroller_pg_password=<set your own>
```

Network ports and protocols

Red Hat Ansible Automation Platform uses several ports to communicate with its services. These ports must be open and available for incoming connections to the Red Hat Ansible Automation Platform server in order for it to work.

Network ports and protocols diagram

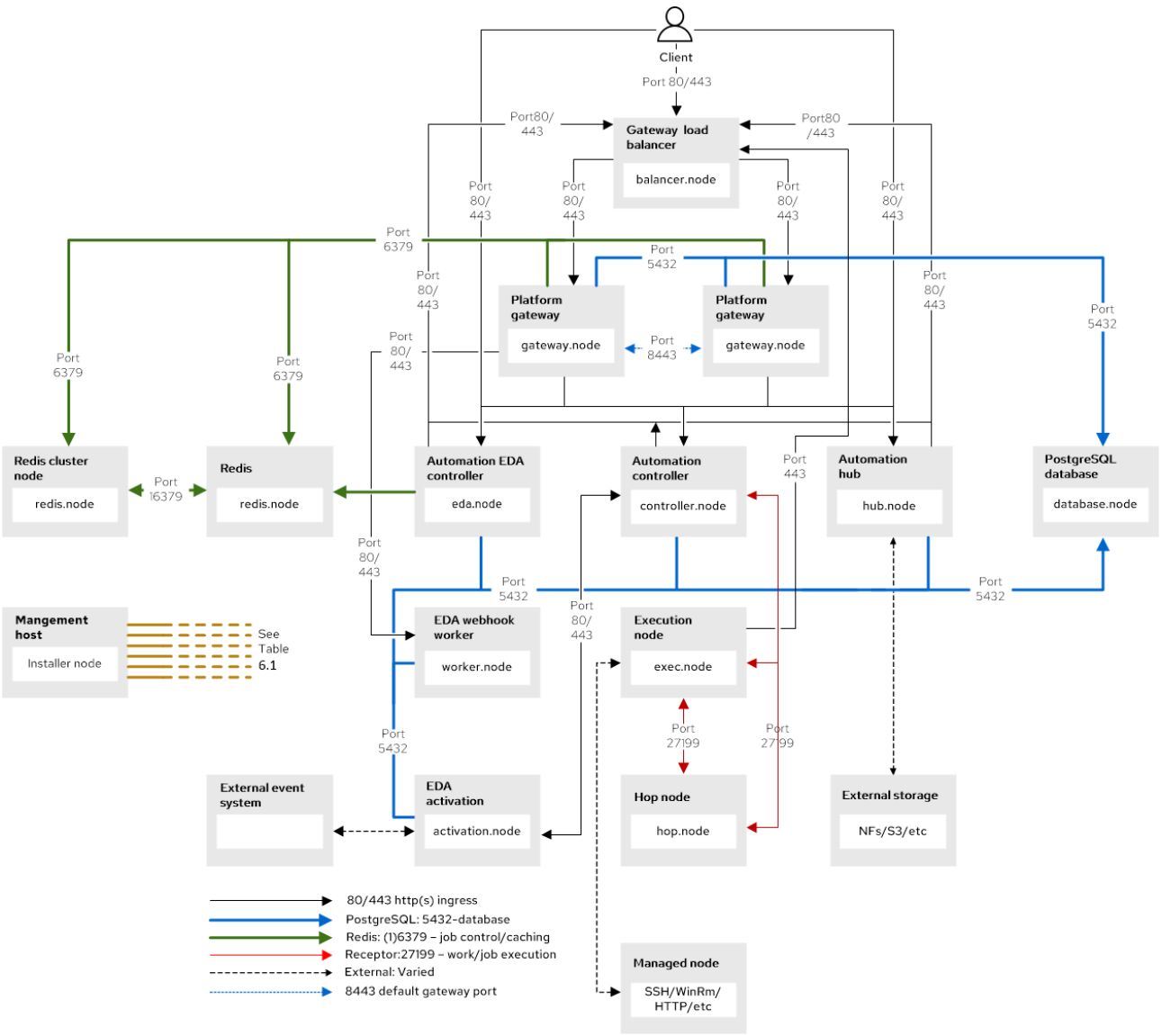
The following architectural diagrams are example of a fully deployed Ansible Automation Platform with all possible components.

NOTE:
In some of the following use cases, hop nodes are used instead of a direct link from an execution node. Hop nodes are an option for connecting control and execution nodes. Hop nodes use minimal CPU and memory, so vertically scaling hop nodes does not impact system capacity.

RPM based installations

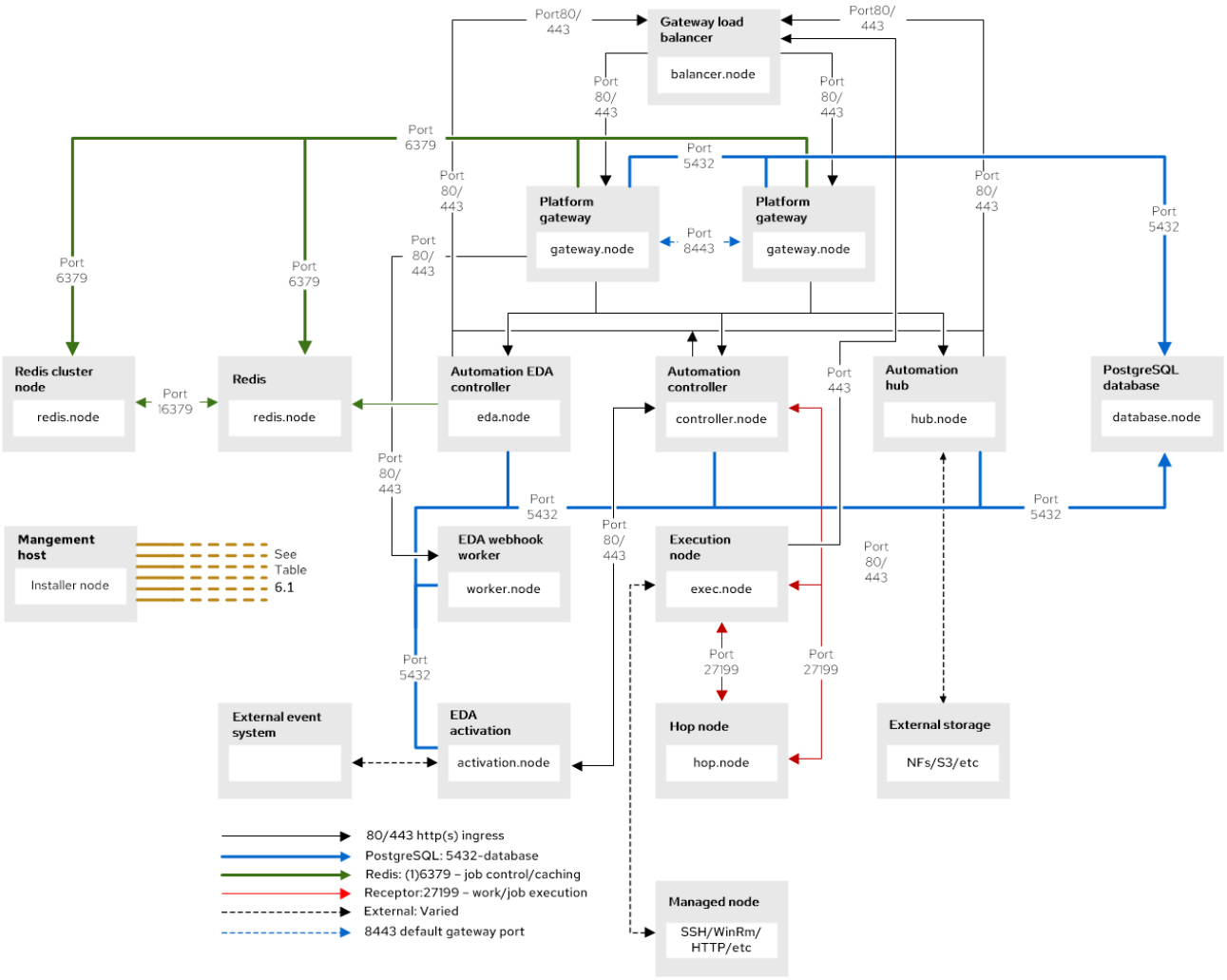
NOTE:
The following diagram shows client initiated connections between Ansible Automation Platform components. Direct connections shown in the diagram between the Client and automation hub, Event-Driven Ansible, and automation controller only apply when systems are upgraded from Red Hat Ansible Automation Platform 2.4 to Red Hat Ansible Automation Platform 2.6. This provides backward compatibility.

Ansible Automation Platform Client initiated network ports and protocols



NOTE:
The following diagram shows internally initiated connections between Ansible Automation Platform components for new installs Red Hat Ansible Automation Platform 2.6.

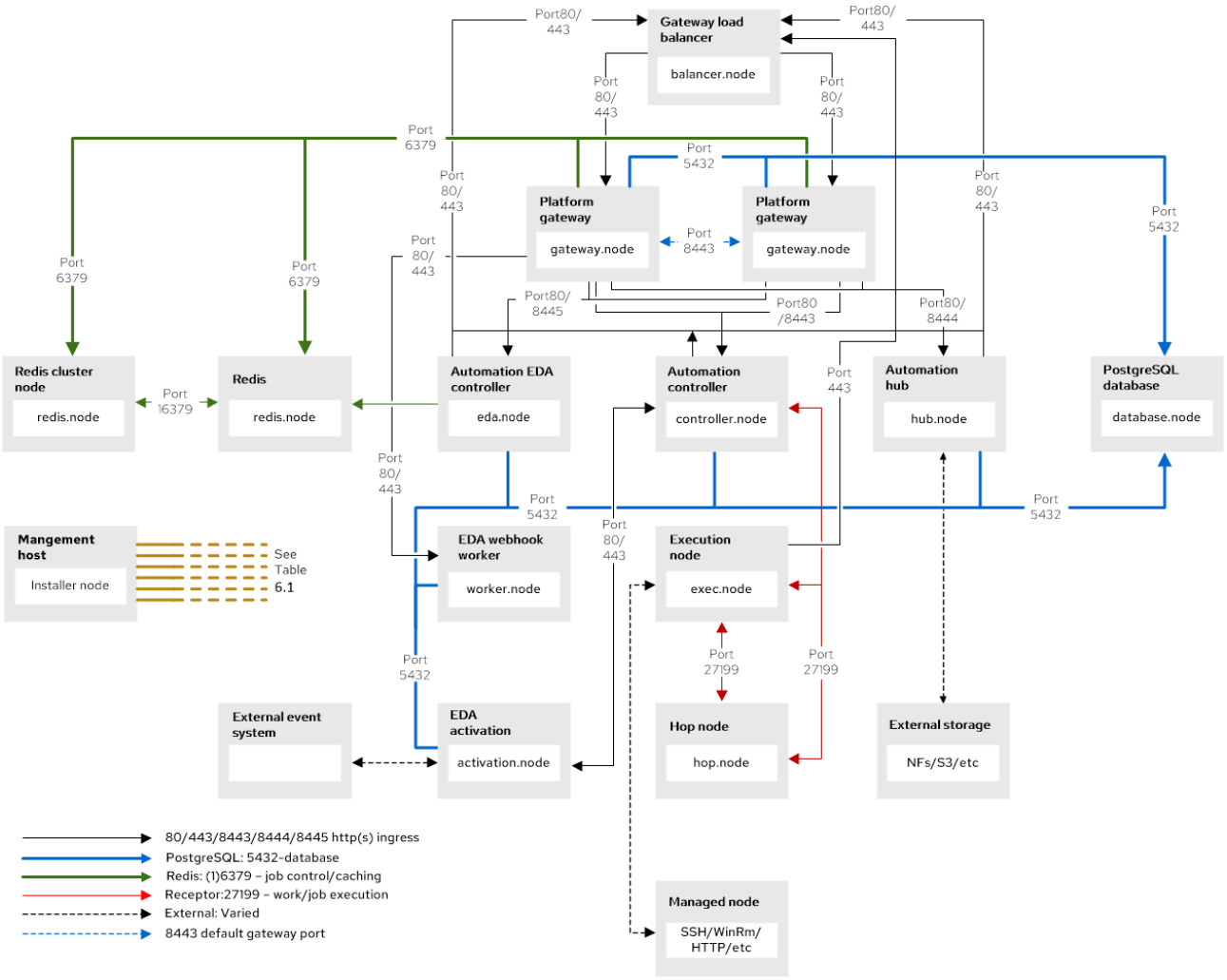
Ansible Automation Platform Internally initiated network ports and protocols



Container-based installations

NOTE:
The following diagram shows connections between Ansible Automation Platform components for a container-based installation Red Hat Ansible Automation Platform 2.6.

Containerized Ansible Automation Platform network ports and protocols



Network ports and protocols table

The following table indicates the destination port and the direction of network traffic:

NOTE:

- The following default destination ports and installer inventory listed are configurable. If you choose to configure them to suit your environment, you might experience a change in behavior.
- Port 443 is the industry standard for HTTPS. Port 80 is not mandatory, but is included for environments that might want to have an unsecure connection.

For RPM-based installations

- Use Port 80 if you set any of `nginx_disable_https`, `automationhub_disable_https` or `automationedacontroller_disable_https` to `true`. See Security-relevant variables in the installation inventory

For container-based installations

- Use Port 80 if you set any of `controller_nginx_disable_https`, `hub_nginx_disable_https` or `eda_nginx_disable_https` to `true`. See Security-relevant variables in the installation inventory

The following table shows container-based installation ports and inventory variables in **bold** text.

Network ports and protocols

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
Automation hub	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	ansible_port
Automation hub	80/443	Installer node	TCP	HTTP/HTTPS	Enables installer node to push the execution environment image to automation hub when using the bundle installer.	ansible_port
Automation hub	80/443	Automation controller	TCP	HTTP/HTTPS	Pull collections	

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
Automation hub	80/443	Event-Driven Ansible node	TCP	HTTP/HTTPS	Pull container decision environments	
Automation hub	80/443	Execution node	TCP	HTTP/HTTPS	Allows execution nodes to pull the execution environment image from automation hub	
Automation hub	80/443	Gateway load balancer/ Ingress node	TCP	HTTP/HTTPS	Accessing the component directly from platform gateway	automationgateway_main_url **gateway_main_url**
Automation hub	443 **8444**	Platform gateway	TCP	HTTPS	Link between platform gateway and Ansible Automation Platform components	
Automation hub	6379	Event-Driven Ansible	TCP	Redis	Event processing	
Automation controller	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	ansible_port
Automation cont	80/443	Event-Driven Ansible	TCP	HTTP/HTTPS	Launch automation controller jobs	

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
roller						
Automation controller	80/443 **80/8443**	Platform gateway	TCP	HTTP/HTTPS	Link between platform gateway and Ansible Automation Platform components	
Automation controller	80/443	Gateway load balancer/ Ingress node	TCP	HTTP/HTTPS	Accessing the component directly from Platform gateway	
Automation controller	27199	Execution node	TCP	Receptor	Used for Mesh peering and communication. See Defining automation mesh node types.	receptor_listener_port peers **receptor_port** **receptor_peers**
Event-Driven Ansible	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	ansible_port
Event-Driven Ansible	80/443 **80/8445**	Platform gateway	TCP	HTTP/HTTPS	Link between platform gateway and Ansible Automation Platform components	
Event-Driven	80/443	Gateway load balancer/	TCP	HTTP/HTTPS	Accessing the component directly from	automationgateway_main_url **gateway_main_url**

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
Ansible		Ingress node			platform gateway	
Event-Driven Ansible	80/443 **8443**	Platform gateway	TCP	HTTPS	Receiving event stream traffic	
Execution node	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	ansible_port
Execution node	443	Gateway load balancer/ Ingress node	TCP	HTTPS		automationgateway_main_url **gateway_main_url**
Execution node	27199	Automation controller	TCP	Receptor	Used for Mesh peering and communication. See Defining automation mesh node types.	receptor_listener_port peers **receptor_port** **receptor_peers**
Execution node	27199	OpenShift Container Platform	TCP	Receptor		
Hop node	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	ansible_port
Hop node	27199	Automation	TCP	Receptor	ENABLE connections from hop	receptor_listener_port

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
		controller			nodes to Receptor port if relayed through hop nodes. See Defining automation mesh node types.	peers **receptor_port** **receptor_peers**
Hop node	27199	Execution node	TCP	Receptor	Used for Mesh peering and communication. See Defining automation mesh node types.	receptor_listener_port peers **receptor_port** **receptor_peers**
Hybrid node	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	`ansible_port`
Hybrid node	27199	Automation controller	TCP	Receptor	ENABLE connections from automation controller to Receptor port if relayed through non-hop connected nodes. See Defining automation mesh node types.	receptor_listener_port peers **receptor_port** **receptor_peers**
PostgreSQL database	22	Installer node	TCP	SSH	Management (Install, Configure, Upgrade)	`pg_port`

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
PostgreSQL database	5432	Automation controller	TCP	PostgreSQL	Required only if the internal database is used with another component. Otherwise, this port should not be open.	automationcontroller_pg_port **controller_pg_port**
PostgreSQL database	5432	Event-Driven Ansible	TCP	PostgreSQL	Required only if the internal database is used with another component. Otherwise, this port should not be open.	automationedacontroller_pg_port **eda_pg_port**
PostgreSQL	5432	Automation hub	TCP	PostgreSQL	Required only if the internal database is used with another component. Otherwise, this port should not be open	automationhub_pg_port **hub_pg_port**
OpenShift Container Platform (RPM only)	6443	Automation controller	TCP	HTTP/HTTPS	Only required when using container groups to run jobs.	Hostname of OpenShift API server
Redis	6379	Automation	TCP	Redis	Job launching	

Destination	Port	Source	Protocol	Service	Required for	Installer Inventory Variable
node		controller				
Redis node	6379	Event-Driven Ansible	TCP	Redis	Job launching	
Redis node	6379	Automation hub	TCP	Redis	Job launching	
Redis node	6379	Platform gateway	TCP	Redis	Data storage and retrieval	
Redis node	16379	Redis node	TCP	Redis	Redis cluster bus port for a resilient Redis configuration	
Mesh ingress	443	Execution node	Receptor	HTTPS	If using mesh ingress, ensure that outbound HTTPS (port 443) is allowed from the execution nodes to the OpenShift route URL.	
Platform gateway	80/443 **80/8444**	Automation hub	TCP	HTTPS	Link between platform gateway and Ansible Automation Platform components	
Platform gateway	8443	Platform gateway	TCP	HTTPS	nginx	

NOTE:

- Hybrid nodes act as a combination of control and execution nodes, and therefore Hybrid nodes share the connections of both.
- If `receptor_listener_port` is defined, the machine also requires an available open port on which to establish inbound TCP connections, for example, 27199.

Network ports and protocols firewalls

The following tables provide information about configuring firewalls for Red Hat Ansible Automation Platform components.

Red Hat Insights for Red Hat Ansible Automation Platform

URL	Required for
https://api.access.redhat.com:443	General account services, subscriptions
https://cert-api.access.redhat.com:443	Insights data upload
https://cert.console.redhat.com:443	Inventory upload and Cloud Connector connection
https://console.redhat.com:443	Access to Insights dashboard

Automation Hub

URL	Required for
https://console.redhat.com:443	General account services, subscriptions
https://catalog.redhat.com:443	Indexing execution environments
https://sso.redhat.com:443	TCP
https://automation-hub-prd.s3.amazonaws.com, https://automation-hub-prd.s3.us-east-2.amazonaws.com	Firewall access
https://galaxy.ansible.com:443	Ansible Community curated Ansible content
https://ansible-galaxy-ng.s3.dualstack.us-east-1.amazonaws.com	Dual Stack IPv6 endpoint for Community curated Ansible content repository

URL	Required for
https://registry.redhat.io:443	Access to container images provided by Red Hat and partners
https://cert.console.redhat.com:443	Red Hat and partner curated Ansible Collections

Execution Environments (EE)

URL	Required for
https://registry.redhat.io:443	Access to container images provided by Red Hat and partners
`cdn.quay.io:443`	Access to container images provided by Red Hat and partners
`cdn01.quay.io:443`	Access to container images provided by Red Hat and partners
`cdn02.quay.io:443`	Access to container images provided by Red Hat and partners
`cdn03.quay.io:443`	Access to container images provided by Red Hat and partners

IMPORTANT:
As of **April 1st, 2025**, `quay.io` is adding three additional endpoints. As a result, customers must adjust allow/block lists within their firewall systems lists to include the following endpoints:

- `cdn04.quay.io`
- `cdn05.quay.io`
- `cdn06.quay.io`

To avoid problems pulling container images, customers must allow outbound TCP connections (ports 80 and 443) to the following hostnames:

- `cdn.quay.io`
- `cdn01.quay.io`
- `cdn02.quay.io`
- `cdn03.quay.io`
- `cdn04.quay.io`
- `cdn05.quay.io`
- `cdn06.quay.io`

This change should be made to any firewall configuration that specifically enables outbound connections to `registry.redhat.io` or `registry.access.redhat.com`.

Use the hostnames instead of IP addresses when configuring firewall rules.

After making this change, you can continue to pull images from `registry.redhat.io` or `registry.access.redhat.com`. You do not require a `quay.io` login, or need to interact with the `quay.io` registry directly in any way to continue pulling Red Hat container images.

For more information, see Firewall changes for container image pulls 2024/2025.

Automation mesh node requirements

Automation mesh is an overlay network that distributes automation work across large and distributed collections of workers through peer-to-peer node connections. Included are the tested system configurations and network port requirements for mesh nodes.

Tested system configurations

Each automation mesh VM has been tested with these requirements: 16 GB RAM, 4 CPUs, 60 GB local disk, and 3000 IOPS.

Network ports

Automation mesh uses several ports to communicate with its services. These ports must be open and available for Red Hat Ansible Automation Platform to work. Ensure that these ports are available and are not blocked by a firewall.

Network ports and protocols

Port number	Protocol	Service	Source	Destination
80/443	HTTP/HTTPS	Receptor	Execution node	OpenShift Container Platform mesh ingress
80/443	HTTP/HTTPS	Receptor	Hop node	OpenShift Container Platform mesh ingress
27199	TCP	Receptor	OpenShift Container Platform cluster	Execution node
27199	TCP	Receptor	OpenShift Container Platform cluster	Hop node

Related information
Scale automation across your infrastructure with automation mesh

Attach your Ansible Automation Platform subscription

You **must** have valid subscriptions on all nodes before installing Red Hat Ansible Automation Platform.

NOTE:
Simple Content Access (SCA) is now the default subscription method for all Red Hat accounts. With SCA, you must register your systems to Red Hat Subscription Management (RHSM) or Satellite to access content. Traditional pool-based subscription attachment commands (such as `subscription-manager attach --pool` or `subscription-manager attach --auto`) are no longer required. For more information, see Simple Content Access.

Procedure

1. Register your system with Red Hat Subscription Management:

```
 $ sudo subscription-manager register --username <$INSERT_USERNAME_HERE> --
password <$INSERT_PASSWORD_HERE>
```

With Simple Content Access (SCA), registration is the only step required to access Ansible Automation Platform content.

> **NOTE:**
> For accounts still using legacy subscription pools, you might have to manually attach subscriptions using the commands shown in the troubleshooting section.

Result

1. Refresh the subscription information on your system:

```
$ sudo subscription-manager refresh
```

2. Verify your registration:

```
$ sudo subscription-manager identity
```

This command displays your system identity, name, organization name, and organization ID, confirming successful registration.

- For legacy accounts not using SCA, you might have to manually attach subscriptions:

```
$ sudo subscription-manager list --available --all | grep -A 30 "Ansible
Automation Platform"
```

This command displays the subscription details including the Pool ID. Look for the `Pool ID:` line in the output.

Once you have identified the correct Pool ID, attach the subscription:

```
$ sudo subscription-manager attach --pool=<pool_id>
```

> **NOTE:**
> Do not use MCT4022 as a `pool_id` as it can cause subscription attachment to fail.

- For legacy accounts not using SCA, if you are unable to locate certain packages that came bundled with the Ansible Automation Platform installer, or if you are seeing a `*Repositories disabled by configuration*` message, use the following steps to identify and enable the required repository:

  a. List available repositories:

  ```
  $ sudo subscription-manager repos --list | grep -i ansible-automation-platform
  ```

  b. Identify the repository name that matches your RHEL version, Ansible Automation Platform version, and architecture (for example, `ansible-automation-platform-2.6-for-rhel-9-x86_64-rpms`).

  c. Enable the repository:

  ```
  $ sudo subscription-manager repos --enable <repository_name>
  ```

Subscription Types

Red Hat Ansible Automation Platform is provided at various levels of support and number of machines as an annual subscription.

- **Standard:**
  - Manage any size environment
  - Enterprise 8x5 support and SLA
  - Maintenance and upgrades included
  - Review the SLA at Product Support Terms of Service
  - Review the Red Hat Support Severity Level Definitions
- **Premium:**
  - Manage any size environment, including mission-critical environments
  - Premium 24x7 support and SLA
  - Maintenance and upgrades included
  - Review the SLA at Product Support Terms of Service
  - Review the Red Hat Support Severity Level Definitions

All subscription levels include regular updates and releases of automation controller, Ansible, and any other components of the Ansible Automation Platform.

For more information, contact Ansible through the Red Hat Customer Portal or at the Ansible site.

Red Hat product documentation legal notices

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If

such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions. "This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code. The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component,

or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions. All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law. No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies. You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions. You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so. A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms. You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms. "Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination. You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies. You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients. Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents. A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom. If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License. Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License. The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES

SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16. If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see link:https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read link:https://www.gnu.org/licenses/why-not-lgpl.html.

Apache license

Version 2.0, January 2004

http://www.apache.org/licenses/

Terms and Conditions for use, reproduction, and distribution

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **"control"** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or **"Your"**) shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **"submitted"** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **"Not a Contribution."**

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a **"NOTICE"** text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications,

or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

www.redhat.com