Ansible Automation Platform

Installation Planning

A trainer-ready, enterprise-grade guide to planning AAP deployments — covering sizing, networking, database design, clustering, and license compliance.

ENTERPRISE ARCHITECTURE | PRE-INSTALL PLANNING | RED HAT AAP

What to Plan *Before* You Install AAP in Production

Successful AAP deployments are won or lost in the planning phase. Before a single package is installed, architects and SREs must align on sizing, topology, database strategy, deployment model, and subscription compliance. This guide walks through every critical decision point.

1. **System Requirements & Sizing**
2. **Network Topology & Ports**
3. **PostgreSQL Backend**
4. **Standalone vs. Clustered**
5. **Licensing & Subscriptions**

Chapter 1

System Requirements & Sizing Guidelines

AAP is **not a monolithic service**. Each component carries distinct compute, memory, and storage demands — and must be sized independently for production workloads.

Controller

Web UI, REST API, scheduling engine, and RBAC. The primary user-facing component.

Execution Nodes

Isolated workers that run Ansible jobs. Scale out horizontally to increase throughput.

Automation Hub

Private repository for Ansible Collections and certified content distribution.

PostgreSQL

Stores all metadata, job state, credentials, RBAC, and audit logs.

EDA Controller

Event-Driven Automation — processes rulebooks and reacts to event sources in real time.

Minimum vs. Production Sizing

LAB / POC — NON-HA

Node	vCPU	RAM	Disk
Controller	4	16 GB	40 GB
Execution Node	2	8 GB	20 GB
PostgreSQL	2	8 GB	50 GB

PRODUCTION — HA RECOMMENDED

Role	vCPU	RAM	Disk
Controller ×2–3	8	32 GB	100 GB
Execution Nodes	4–8	16–32 GB	50 GB
Automation Hub	4	16 GB	200+ GB
PostgreSQL (HA)	8	32 GB SSD	200+ GB

Key Sizing Drivers

These variables determine your actual hardware footprint more than any reference architecture:

Parallel Jobs

Peak concurrent job forks drive CPU and memory demand on execution nodes.

Inventory Scale

Large dynamic inventories increase DB query load and Controller memory pressure.

Playbook Complexity

Long-running plays with many tasks extend execution node CPU utilization windows.

Artifact Retention

Job stdout and event logs accumulate rapidly — size PostgreSQL disk accordingly.

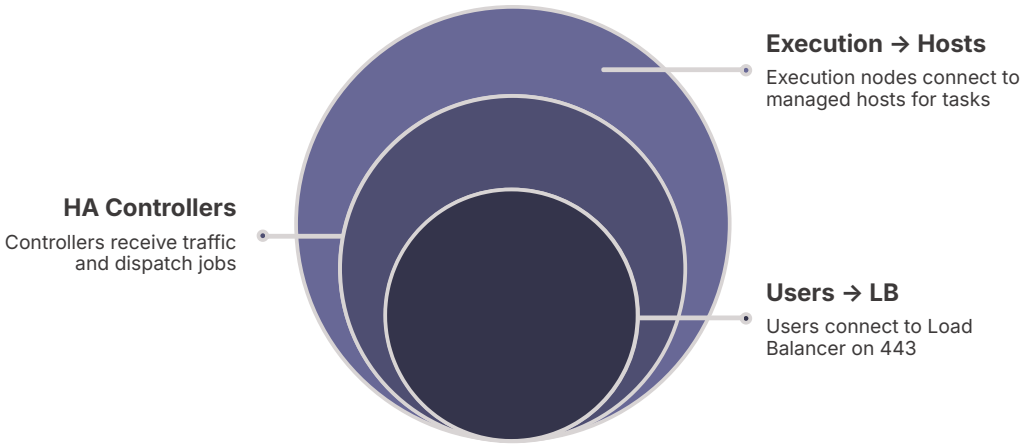
EDA Event Load

High-frequency event sources (Kafka, webhooks) demand dedicated EDA node resources.

Chapter 2

Network Topology & Port Requirements

A well-designed network topology is foundational to a resilient AAP deployment. Controllers must be reachable by users, able to dispatch jobs to execution nodes, and isolated from direct external exposure at the database tier.



Design Principles

Load Balancer Required

All user and API traffic must enter via a load balancer — never directly to a single Controller node.

Internal-Only Tiers

Automation Hub and PostgreSQL should have no external exposure. Restrict to Controller network segment.

Segmented Execution

Execution nodes operate outbound-only to managed hosts — they never accept inbound connections from targets.

Critical Ports — Must Be Open

⚠️ Firewall misconfiguration is the **#1 cause of failed AAP installs**. Validate every path before running the installer.

Source	Destination	Port	Purpose
User / Browser	Controller	443	Web UI & REST API access
Controller	Execution Nodes	27199	Job dispatch & receptor mesh
Controller	PostgreSQL	5432	Database reads/writes
Controller	Automation Hub	443	Collection sync & downloads
Execution Node	Managed Hosts	22 / 5986	SSH (Linux) / WinRM (Windows)
EDA Controller	Event Sources	Custom	Kafka topics, webhooks, alerts

Run `nc` or `telnet` connectivity tests from each source node **before** executing the installer. Document results as part of your pre-install checklist.

Chapter 3

Database Backend — PostgreSQL Basics

What PostgreSQL Stores

Job Metadata

Run history, status, timing, and output references.

Inventories

Host groups, variables, and dynamic inventory cache.

Credentials

Encrypted secrets for SSH, cloud, vault, and SCM.

Schedules & RBAC

Job schedules, role assignments, and permission mappings.

Audit Logs

Full activity trail for compliance and forensics.

What PostgreSQL Does *Not* Store

❌ Playbooks

Stored in SCM (Git). Controller clones at job runtime.

❌ Collections

Stored on Automation Hub filesystem or Execution Node cache.

❌ Host State

AAP is stateless toward managed hosts — no configuration drift data stored natively.

Database Deployment Options & Best Practices

Deployment Models

1

Bundled PostgreSQL

Installed by the AAP installer on the Controller node. Suitable for **Lab / POC** only — no HA, single point of failure.

2

External PostgreSQL

Dedicated DB host managed independently. Required for **production** deployments to enable separate scaling and maintenance windows.

3

HA PostgreSQL (Patroni)

Streaming replication with automatic failover. Required for **mission-critical** environments with zero-downtime SLAs.

Production Best Practices

SSD storage only

Spinning disks cause severe job latency under concurrent write loads.

Isolate DB from Controllers

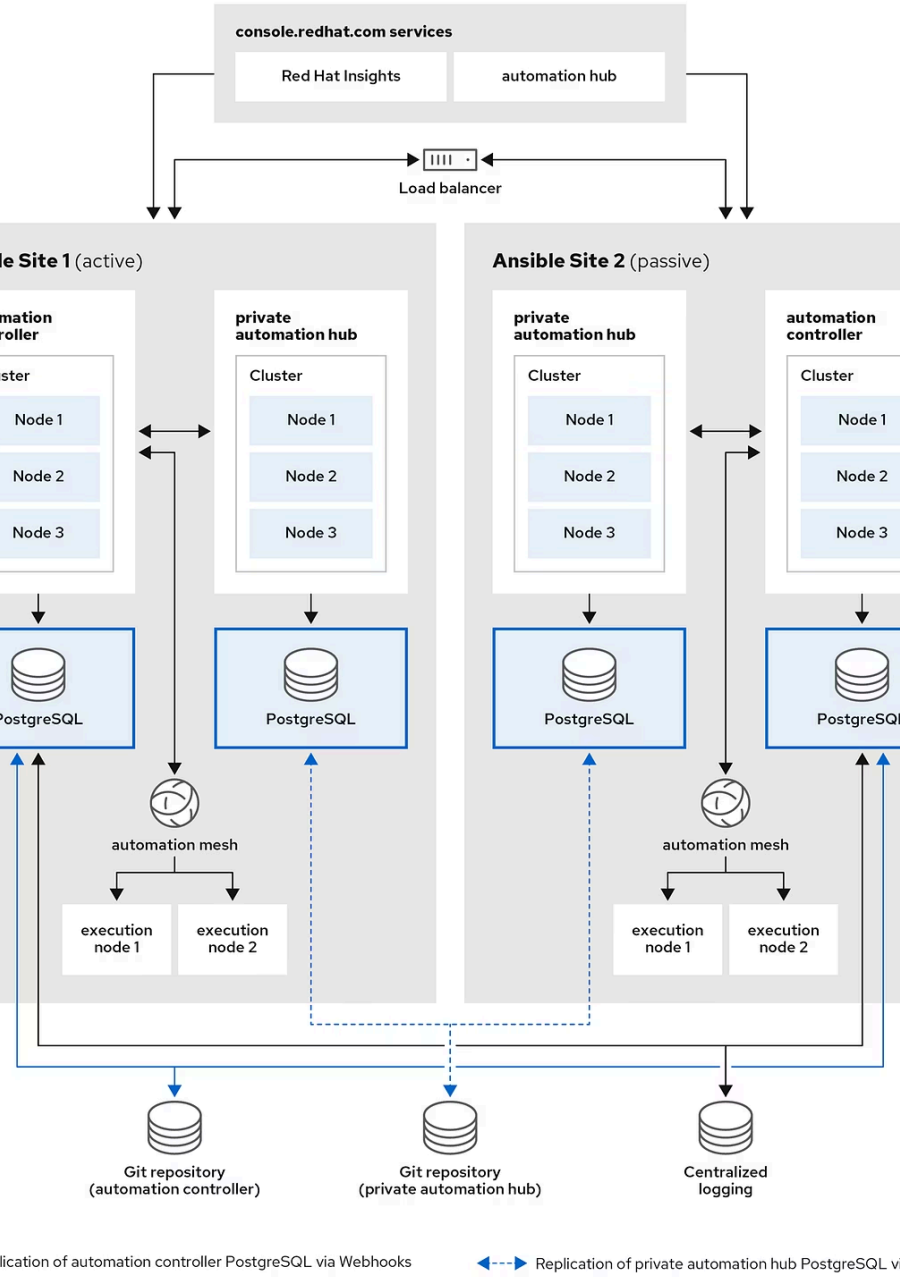
Never co-locate PostgreSQL with Controller — resource contention degrades both.

Daily backups + WAL archiving

Enable point-in-time recovery. Test restores regularly — untested backups are not backups.

Dedicated DB instance

Do not share the PostgreSQL instance with other applications. Schema conflicts and connection limits will cause production incidents.



Chapter 4

Installation Types: Standalone vs. Clustered

Choosing the right deployment model determines your availability profile, scalability ceiling, and operational complexity from day one.

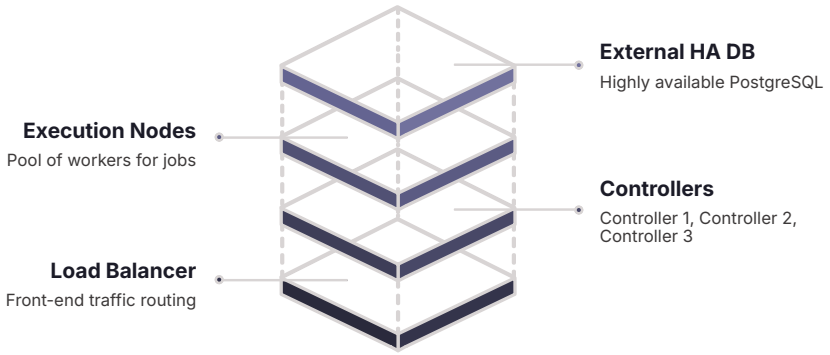
Standalone

Controller + Execution + DB on a single VM. Fast to deploy, zero HA. Ideal for training, demos, and PoC environments.

Clustered

Multiple Controllers behind a load balancer, dedicated execution nodes, and external HA PostgreSQL. Production-ready with horizontal scale.

Clustered Installation — Architecture Deep Dive



Why Clustered Is Recommended

High Availability

Controller failure is absorbed by remaining nodes — user sessions and running jobs are unaffected.

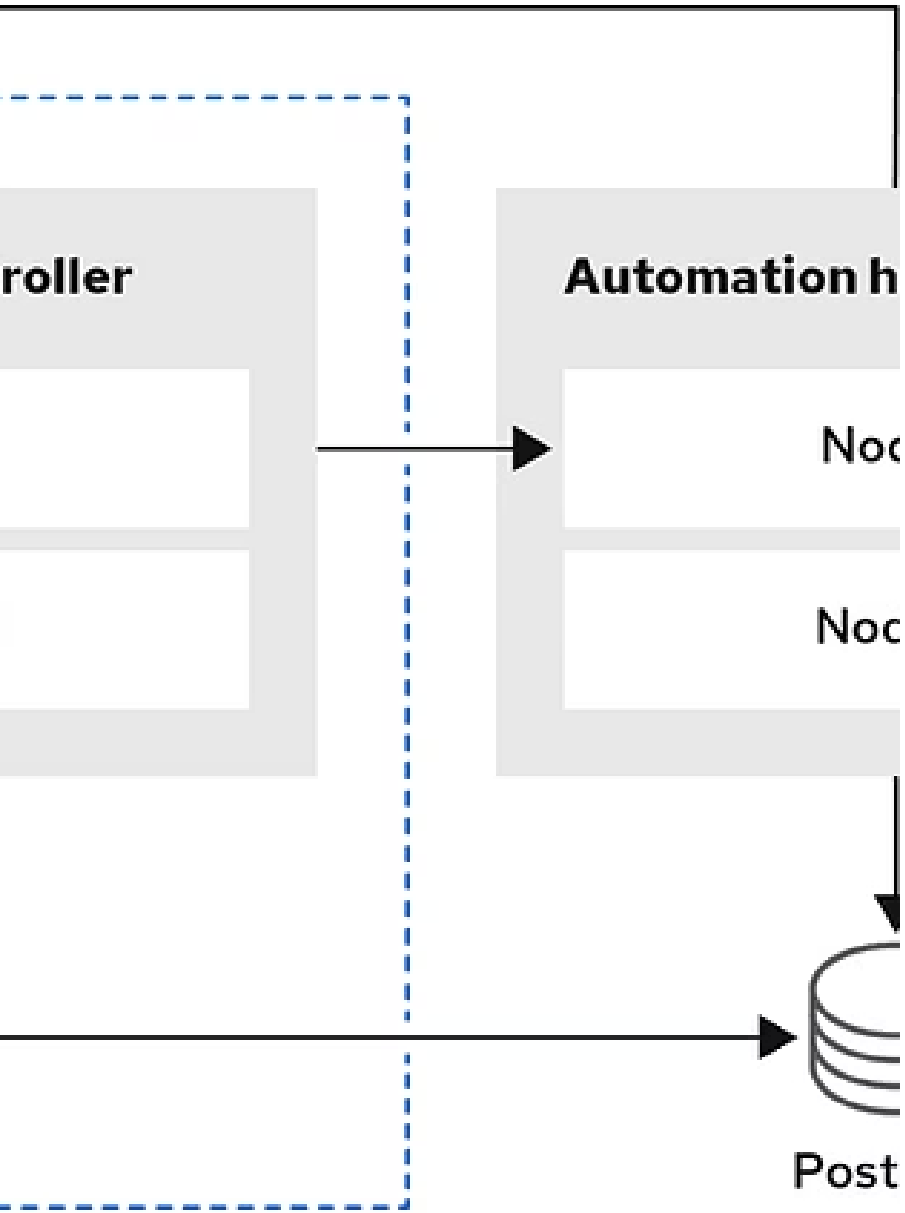
Horizontal Scaling

Add execution nodes independently as job throughput demands grow — no Controller changes required.

Zero-Downtime Upgrades

Rolling upgrades across Controller nodes eliminate maintenance windows.

📌 **Controllers are stateless** — all state lives in PostgreSQL. This is what makes horizontal scaling safe and predictable.



Chapter 5

Licensing Models & Subscription Management

AAP licensing is consumption-based. The unit of measure is the **managed node** — any host automated by Ansible at least once within a rolling 24-hour window, regardless of OS or platform type.

Linux Hosts

Physical, virtual, or containerized Linux targets each count as one managed node.

Windows Hosts

WinRM or SSH-managed Windows servers consume licenses at the same per-node rate.

Network Devices

Routers, switches, and firewalls managed via network modules count as managed nodes.

Cloud & Containers

Cloud instances and container workloads managed through dynamic inventory are fully counted.

Subscription Types & Use Cases

Standard

Designed for smaller teams and departmental automation. Includes business-hours support with standard SLAs.

Premium

Full enterprise entitlement with 24×7 support, faster response SLAs, and access to Technical Account Management.

Developer

Single-user, non-production license. Ideal for personal labs, training environments, and proof-of-concept builds.

Satellite Add-on

Integrates AAP with Red Hat Satellite for unified lifecycle management, patch orchestration, and content governance.

Choose your subscription tier based on team size, support SLA requirements, and integration with existing Red Hat infrastructure tooling.

Subscription Management in Practice

How Subscriptions Are Managed

01

Register via Red Hat Customer Portal

Attach your subscription manifest online or generate an offline activation key for air-gapped environments.

02

Import Manifest into Controller

Upload the subscription manifest through the Controller UI or API under **Settings → Subscription**.

03

Monitor Usage in Controller UI

Real-time managed node counts are visible in the dashboard. Threshold alerts warn before limits are reached.

Compliance Behavior

Overuse ≠ Immediate Shutdown

AAP does not hard-stop when node counts are exceeded — automation continues running.

Compliance Tracked for Audits

Red Hat tracks consumption data. Overuse is flagged during subscription reviews and true-up cycles.

Air-Gapped Activation Supported

Offline manifest activation is fully supported for disconnected or classified network environments.

Pre-Install

Installation Planning Checklist

Complete every item before executing the AAP installer. Partial preparation is the primary cause of failed or re-run deployments.

1

Hardware Sized

vCPU, RAM, and disk validated against production sizing guidelines for all roles.

2

Network Ports Validated

Connectivity tested end-to-end for all required source → destination port combinations.

3

DNS & NTP Configured

Forward/reverse DNS resolves for all nodes. NTP synchronized across the entire cluster.

4

PostgreSQL Decision Made

Bundled, external, or HA PostgreSQL (Patroni) selected and provisioned.

5

HA vs. Standalone Chosen

Deployment model documented and inventory file prepared accordingly.

6

Subscription Ready

Manifest downloaded from Customer Portal and available for import post-install.

7

Backup Strategy Defined

Daily backup schedule, WAL archiving, and restore procedures documented and tested.

8

Security Review Completed

SELinux, firewall rules, TLS certificates, and credential encryption requirements signed off.

Trainer & Interview Summary

> "Ansible Automation Platform installation planning focuses on **correct component sizing**, **network readiness**, **PostgreSQL design**, choosing between **standalone and clustered deployments**, and ensuring **license compliance** based on managed node consumption."

Size Each Component Independently

Controller, Execution Nodes, Hub, PostgreSQL, and EDA each have distinct resource profiles.

Validate Network Before Install

Every required port must be confirmed open. Firewall gaps are the #1 install failure cause.

Use External HA PostgreSQL in Production

Bundled DB is for labs only. Production requires isolation, SSD, backups, and WAL archiving.

Cluster for Production, Standalone for Labs

Controllers are stateless — clustering is safe, scalable, and required for HA SLAs.

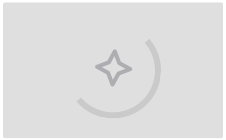
License by Managed Nodes

Any host automated in a 24-hour window counts. Monitor in Controller UI and plan for growth.

What's Next

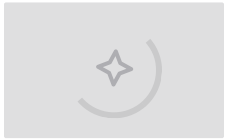
Continue Your AAP Journey

Installation planning is the foundation. The following modules build on this knowledge with hands-on implementation and advanced deployment scenarios.



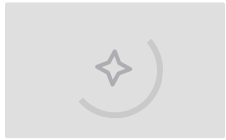
Clustered Install Lab

A step-by-step hands-on lab walking through a full production-grade clustered AAP installation with a live inventory file.



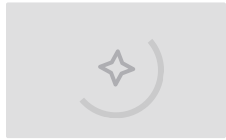
Production Inventory File

A fully annotated, production-ready AAP installer inventory file covering all roles, variables, and HA configuration options.



OpenShift-Based AAP Deployment

Planning and deploying AAP on OpenShift using the AAP Operator — covering resource quotas, storage classes, and GitOps integration.



Trainer Notes & Slides

Formatted trainer notes with talking points, interview Q&A, and slide decks ready for Red Hat design workshops and production planning calls.