**🔍 Ansible Automation Platform (AAP) vs. Ansible Core / AWX**

| **Feature/Capability** | **Ansible Core** | **AWX (Community Edition)** | **Ansible Automation Platform (AAP)** |
| --- | --- | --- | --- |
| **Publisher** | open source community) | Red Hat (community project) | Red Hat (enterprise product) |
| **Support** | Community only | Community only | Full Red Hat support (SLA-based) |
| **Updates** | Frequent, fast | Community-paced | Stable releases, tested, enterprise-grade |
| **Web UI (GUI)** | ❌ Not included | ✅ Yes (AWX UI) | ✅ Yes (Automation Controller) |
| **REST API** | ❌ No | ✅ Yes | ✅ Yes (fully supported + backward-compatible) |
| **RBAC (Role-Based Access Control)** | ❌ No | ✅ Limited | ✅ Advanced and customizable |
| **Job Scheduling** | ❌ No | ✅ Yes | ✅ Yes, with workflow chaining |
| **Workflow Automation** | ❌ No | ✅ Yes | ✅ Yes (GUI + YAML) |
| **Logging & Auditing** | Manual | Basic logs | Centralized, enterprise audit-ready |
| **Secrets Management** | Ansible Vault (CLI) | Basic | Integration with Vaults + RBAC |
| **Scalability** | Single node | Single/multi-node (manual) | Enterprise-scale, clustering, HA |
| **Execution Environments (EEs)** | Manual setup | Basic support | Full support with ansible-builder, registry integration |
| **Content Management** | CLI | Limited | ✅ Automation Hub + Private Repos |
| **CI/CD Integration** | Manual | Basic | ✅ Deep Git integration, webhooks, API |
| **Analytics** | ❌ No | ❌ No | ✅ Automation Analytics (via cloud.redhat.com) |
| **Offline Content Delivery** | ❌ No | ❌ No | ✅ Available via disconnected Automation Hub |
| **License** | GPL | Apache | Requires RHEL subscription and AAP license |
| **Best For** | Individuals, CLI users | Testing GUI workflows | Enterprises, regulated environments |

**📦 What is AWX?**

**AWX** is the **open-source upstream project** of Ansible Tower / Automation Controller. Think of it as the "development version" of AAP’s UI and API features.

* It provides a web interface, REST API, and task engine.
* It is **not supported by Red Hat**, intended for **test/dev environments**.
* Installable via containerized setup using awx-operator.

**💼 What is Ansible Automation Platform (AAP)?**

**AAP** is Red Hat’s **commercial enterprise solution** that:

* Combines Automation Controller (based on AWX),
* Automation Hub,
* Private Galaxy for content,
* Automation Mesh for scaling,
* And Execution Environments for containerized, portable runtimes.

It’s designed for **security, performance, support, and scalability** in production environments.

**🧠 Key Differences: AAP vs AWX/Core**

| **Aspect** | **AWX/Core** | **AAP** |
| --- | --- | --- |
| **Installation** | Manual (docker, operator) | Official installer, containerized or VM |
| **Execution Environments** | Supported, manual | First-class support, integrated |
| **Automation Hub** | Not available | ✅ Use private/custom collections |
| **Analytics** | ❌ | ✅ (Red Hat hosted or disconnected) |
| **Role-Based Access** | Limited | Granular RBAC and orgs/teams |
| **License** | Open Source | Paid subscription |
| **Offline Use** | ❌ | ✅ (disconnected deployments supported) |
| **Security & Compliance** | Manual | FIPS, audit logging, SCAP/CIS automation |

**✅ Enterprise Benefits of AAP**

| **Enterprise Requirement** | **AAP Feature** |
| --- | --- |
| **Security & Compliance** | FIPS-compliant cryptography, SSO integration, RBAC, auditing |
| **Governance** | Role delegation, job approvals, logs |
| **Reliability** | HA, clustering, automation mesh |
| **Scalability** | Job slicing, multi-node controller |
| **Content Curation** | Automation Hub, private Galaxy, signed collections |
| **Performance Optimization** | Execution Environments, custom base images |
| **Support** | 24x7 enterprise support by Red Hat |
| **DevOps Integration** | CI/CD pipelines, webhook triggers, GitOps ready |

**🛠️ Example Use Cases by Platform**

| **Use Case** | **Suitable Platform** |
| --- | --- |
| Test playbooks locally or via CLI | Ansible Core |
| Small team experimenting with GUI | AWX |
| Enterprise production automation | AAP |
| Multi-team RBAC and auditing | AAP |
| Centralized job scheduling | AWX / AAP |
| Disconnected / air-gapped automation | AAP only |

**✅ Visualized Architecture: AAP 2.5 Containerized on Single Node (All-in-One)**

+--------------------------------------------------------------------------------------+

| RHEL 9 Host (rhel9master.ansible.local) |

| [4 vCPU | 16 GB RAM | 60 GB Disk | Podman Installed] |

+--------------------------------------------------------------------------------------+

| |

| +------------------------+ +---------------------+ +----------------------+ |

| | Automation Controller| | Automation Hub | | Event Driven Ansible| |

| | (controller) | | (hub) | | (eda-server) | |

| | - Web UI/API | | - Collections repo | | - WebSocket events | |

| | - Job templates | | - EE signing keys | | - Rulebooks | |

| +----------+-------------+ +----------+----------+ +-----------+----------+ |

| | | | |

| | | | |

| v v v |

| +--------------+ +--------------------+ +------------------+ |

| | PostgreSQL DB |<------>| controller/hub/eda |<------->| Redis (Standalone)| |

| | (Single DB) | | use local database | | For real-time data| |

| +--------------+ +--------------------+ +------------------+ |

| |

| +-------------------------------------------------------------------------------+ |

| | Execution Environments (Podman containers) | |

| | - ansible-core | |

| | - collections | |

| | - python dependencies | |

| +-------------------------------------------------------------------------------+ |

| |

| +---------------------+ +----------------------+ +---------------------+ |

| | Automation Gateway | --> | Authentication Proxy | --> | Internal Routing | |

| | - Single entrypoint | | - SSO/AuthN | | - API coordination | |

| +---------------------+ +----------------------+ +---------------------+ |

| |

+--------------------------------------------------------------------------------------+

<< External Access >>

https://rhel9master.ansible.local

**🔍 Component Roles in Your Deployment**

| **Component** | **Deployed As** | **Function** |
| --- | --- | --- |
| **Automation Controller** | Podman container | Orchestration engine; manages playbooks, inventories, and jobs |
| **Automation Hub** | Podman container | Collection repository; stores certified/validated Ansible content |
| **Event-Driven Ansible** | Podman container | Handles rules + events for automation beyond scheduled triggers |
| **PostgreSQL DB** | Podman container | Backend DB for controller, hub, and EDA |
| **Redis** | Podman container | Event streaming, caching |
| **Execution Environments** | Podman containers | Runs ansible-core with custom dependencies |
| **Automation Gateway** | Podman container | Unified access layer for all platform components |