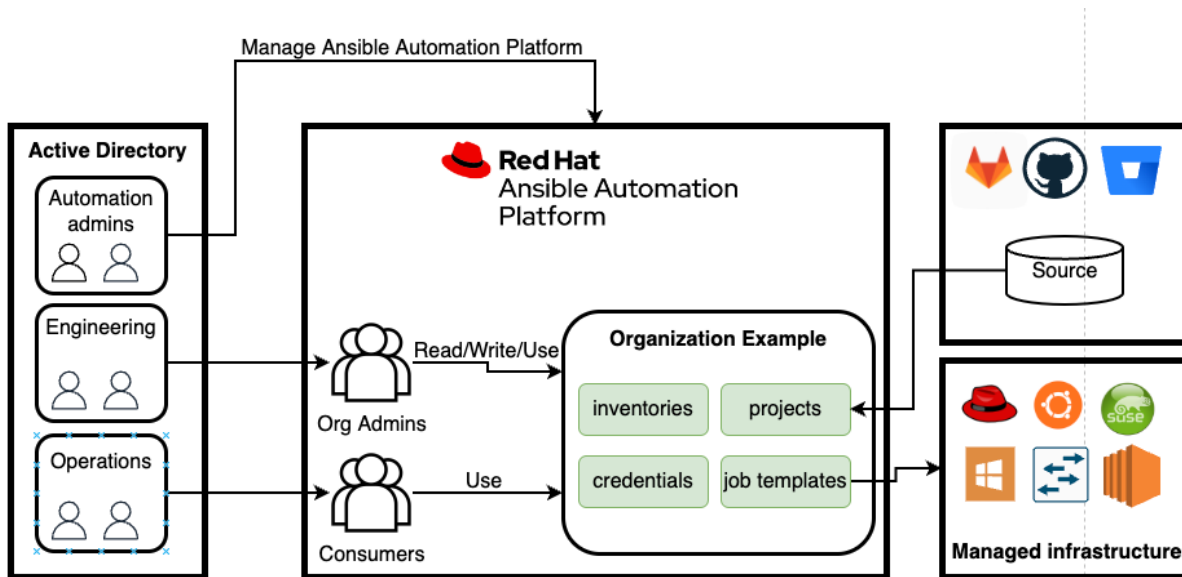


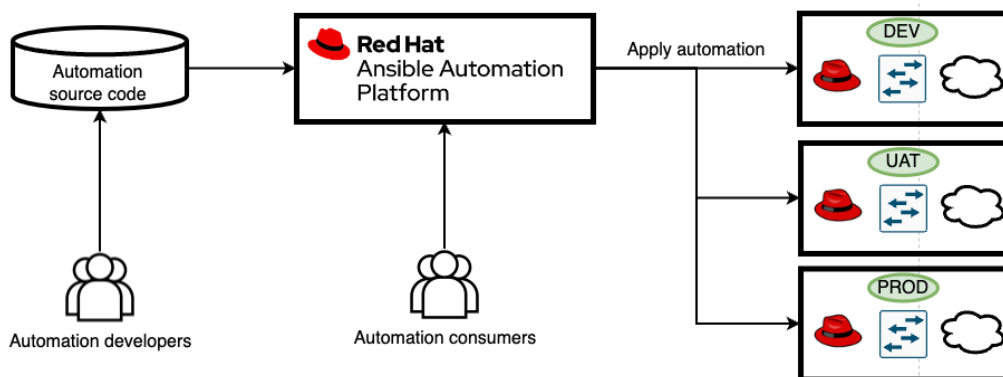
Li9 Technology Solutions Provides customers with an Automation Pilot that focuses on a number of automation use cases across multiple domains to provide consistent and integrated automation across a company's IT Organization.

A company's IT consists of multiple IT domains and infrastructure levels, including compute, virtualization, containerization, storage, networking, and security. All these levels require significant effort from operations, engineering, and DevOps teams.

High Level Architecture



Deliver changes to all environments



This Pilot allows IT organizations to rapidly adopt and evaluate configuration management processes using Ansible technology with a structured and secure strategy.

The Pilot is different from a POC in that it provides specific deliverables, timelines and an implementation that is the base for a production solution.

After the customer's requirements are determined a formal SOW will be created with specific deliverables and success criteria.

Highlights	<ul style="list-style-type: none">● Red Hat Ansible automation pilot● Industry-standard automation platform● Standardization of automation scenarios and processes
Flexibility	<ul style="list-style-type: none">● Work according to your schedule● The service delivery takes between 2-3 weeks
Outcomes	<ul style="list-style-type: none">● Specific deliverables with measurable successes● Ready to use automation infrastructure● Routine tasks are automated● Multiple Customer teams can develop consistent and integrated automation with ansible

Overview

Red Hat Ansible Automation Platform is the foundation for building and operating automation services at scale, providing enterprises a composable, collaborative, and trusted execution environment. It meets customers where they are in their automation journey, bringing them a flexible automation platform to facilitate success from multiple points in the IT infrastructure.

Red Hat Ansible Automation Platform contains multiple components including Ansible Engine and Automation controller.

Ansible Engine is an open source automation tool and it responds to enormous automation requirements such as flexibility, fast speed of delivery, reusability, compatibility. Thanks to its simplicity, it has become a great competitor to solutions such as puppet, chef, salt stack and has grown to a lead tool for automating from small to very large enterprise level environments. It is used as a configuration tool and a management solution for many on prem and cloud services. Ansible helps to implement configuration management, device management, etc.

Enterprises need to manage Ansible at scale with all features like RBAC, AD integration, logging, etc. This is solved by Red Hat Ansible Controller.

Red Hat Ansible Automation Controller is a part of Red Hat Automation Platform that can be used separately. It is supplied with a huge number of free modules and collections of roles, for most causes required in IT from on prem to cloud services. Automation Controller uses Ansible as a backend tool and therefore is compatible with all its modules, plugins, roles and collections. All Automation Controller operations might be performed from RESTful API calls. That allows it to make it a powerful engine of your CI/CD processes and also integrate it in your existing workflows.

Li9 Technology Solutions provides a rapid Ansible Automation implementation service that allows customers to see value from Ansible automation within a short period of time.

Li9 is the technical partner of Red Hat Inc. Li9 focuses on helping customers enable modern IT technologies and the DevOps approach to assist the business in achieving goals faster and move new products into production. Red Hat leverages Li9's expertise in some of their largest accounts to provide Ansible Automation and OPenShift Container solutions.

High Level Pilot Overview

This service is a quick way to have a fully functional Red Hat Ansible Automation Platform, that is ready to serve your automation scenarios.

Planning: Our project team will work with your experts to understand your infrastructure, routine processes, infrastructure stack, development and DevOps

processes, goals and priorities, technical requirements, and automation platform use cases. During this call we have to identify our target use cases to be automated.

Discovery: Our project team will perform a discovery session to understand your environmental architecture, focusing on infrastructure layers, equipment and solutions, such as compute, virtualization, containerization, storage, networking, security and public cloud. We will also collect all the required information to proceed with implementation.

Prerequisites: Our project team will help you to configure service prerequisites. Please check for the full list of prerequisites in the “[Service Prerequisites](#)” chapter. At this stage, we will also validate that all prerequisites are met.

Deployment and Configuration: We will ensure if the Red Hat Ansible Automation Platform is installed and properly configured to provide you its functionality. Please check the “[Deliverables](#)” chapter for details.

Automation Examples: This is the most important part of the service. We will work on automating a few of your simple routine tasks related to your infrastructure. This may include a number of activities including playbook(s) development, inventory creation, credentials creation, project creation, job template creation. This service includes limited number of qualified use cases. Please check the “[Limitations](#)” and “[Automation use cases](#)” chapters for details.

Demonstration: We will perform a demonstration of automation execution and handle Q/A Session with your team and provide hands-on knowledge transfer. A 2-day, 8-hour hands-on Ansible class is also available to customer after the Pilot project is complete.

Solution

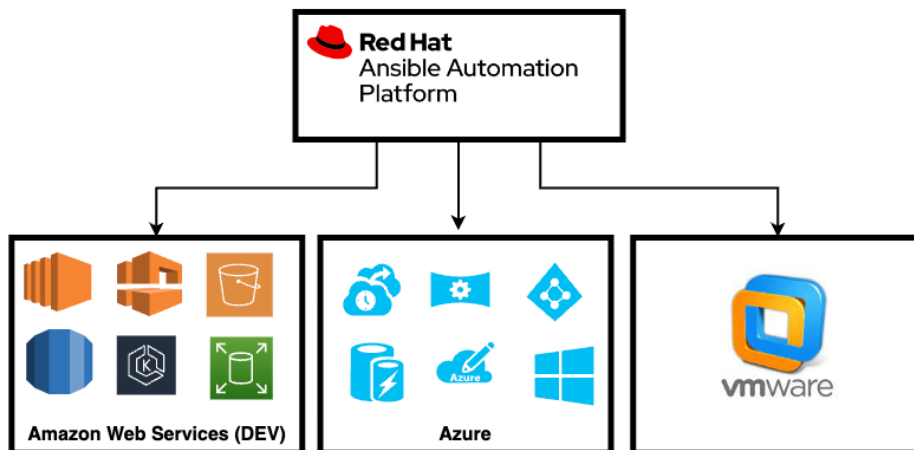
Automation Use Cases

During the planning phase, we have to identify the target automation use case(s). As the part of the service delivery we will implement some basic automation that may help you to move forward faster with Ansible.

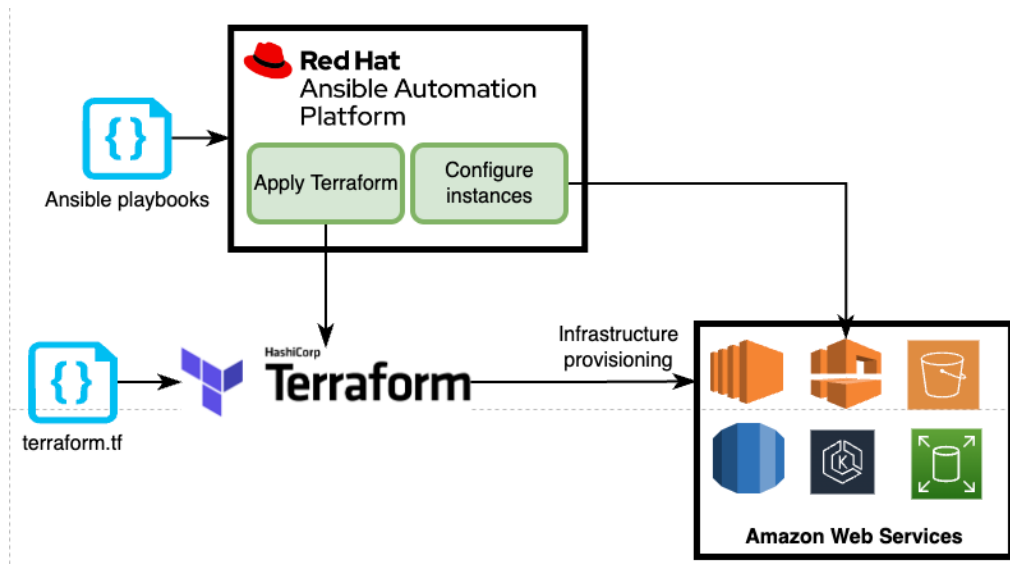
Infrastructure as Code

- Public cloud automation (AWS, Azure, GCP)
- Private cloud automation (VMware, OpenStack)

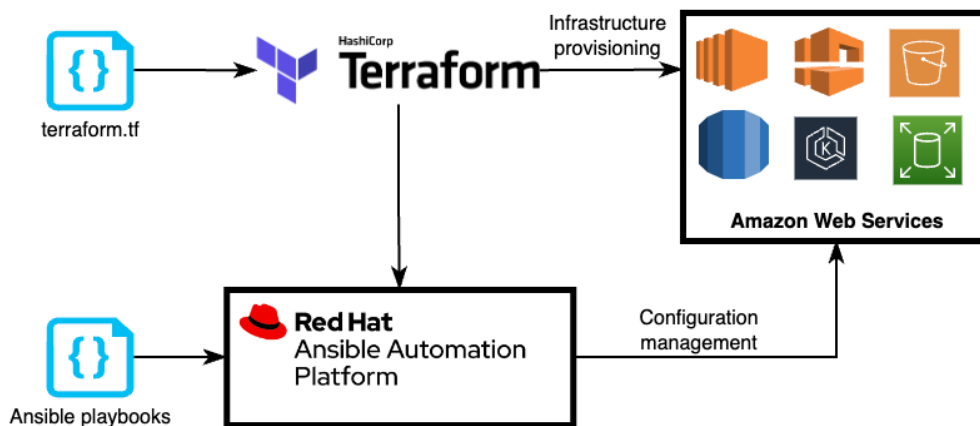
Option 1 - Use Ansible only



Option 2 - Ansible uses Terraform as a backend



Option 3 - Terraform uses ansible for configuration



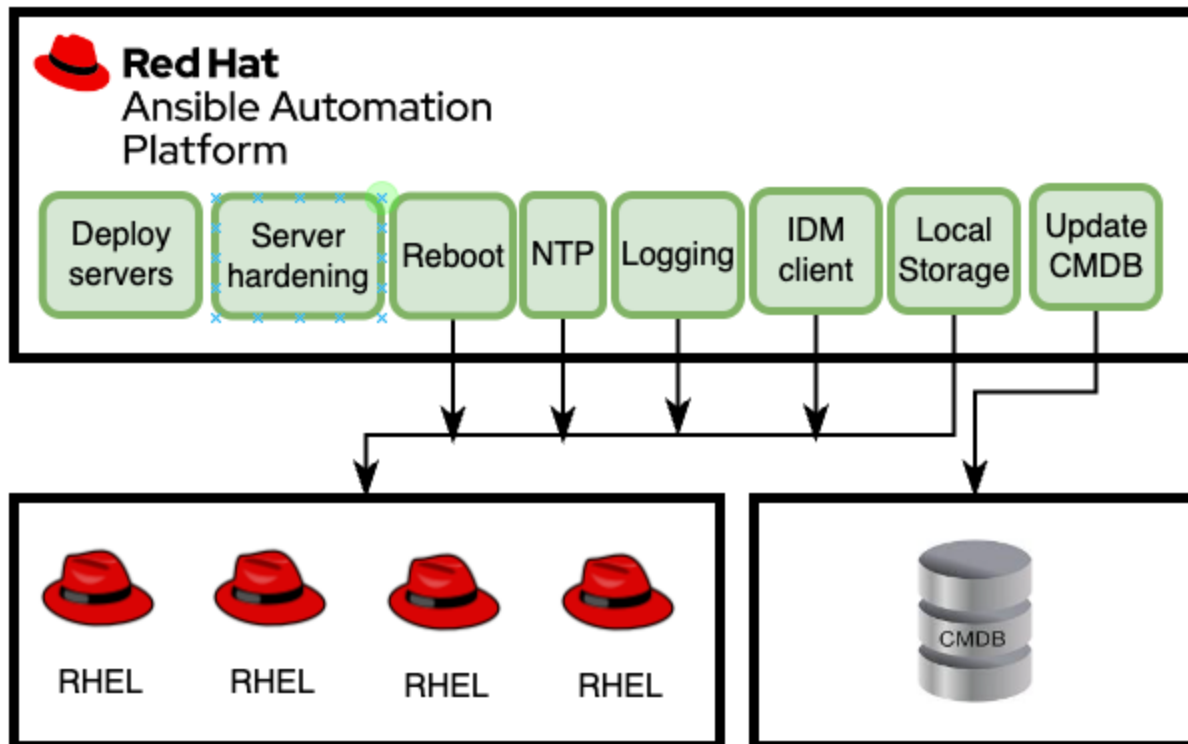
Routine Task Automation

There a number of possible scenarios that fall under this category but here are a few examples:

- Servers
 - Run a command on all servers
 - Manage local users and their passwords
 - Manage local storage

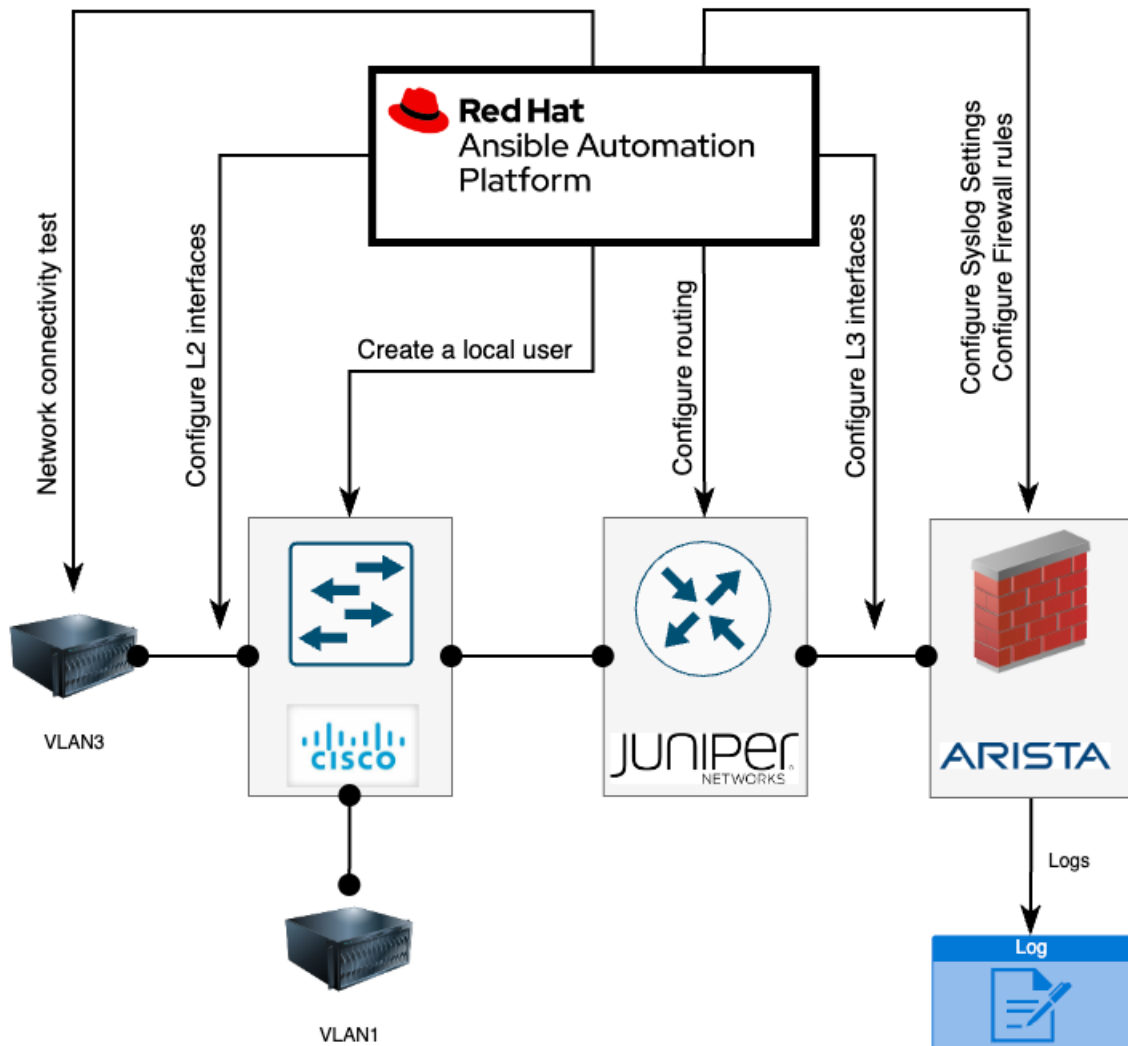
- Server and application restarts
- Report creation
- Network devices
 - Manage ACLs
 - Configure routes
 - Enabling/disabling interfaces
 - Information gathering (like IOS version, etc)
- Public clouds

Server Provisioning



- Linux server post deployment configuration
 - NTP
 - Syslog
 - Access control
- Linux server hardening
- Linux service/application installation and configuration

Network device automation



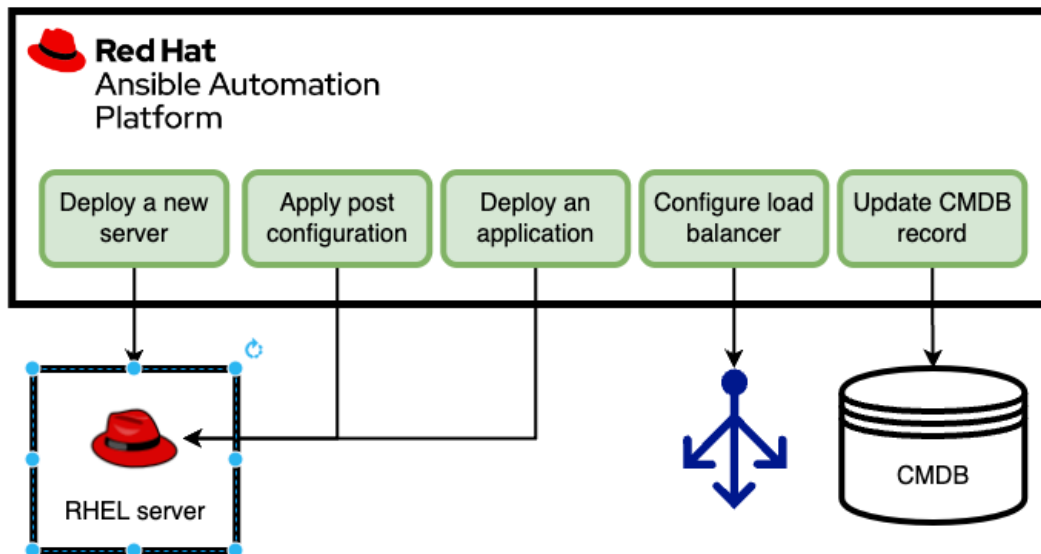
- Multi-Vendor Automation
 - Cisco
 - Juniper
 - Arista
 - F5
- Network Device Configuration Management
 - Gathering device facts and configs
 - SNMP polls/traps
 - NTP settings
 - Local passwords
 - Syslog settings

- ACLs
- L2/L3 Interface settings
- Configure routing

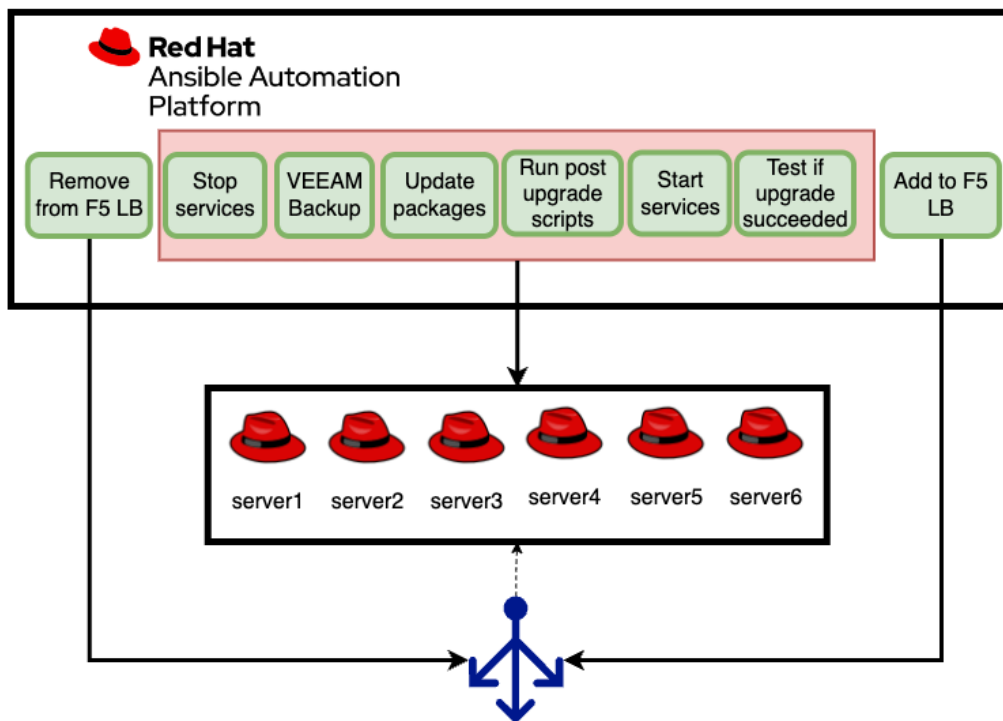
Workflow automation

New server provisioning

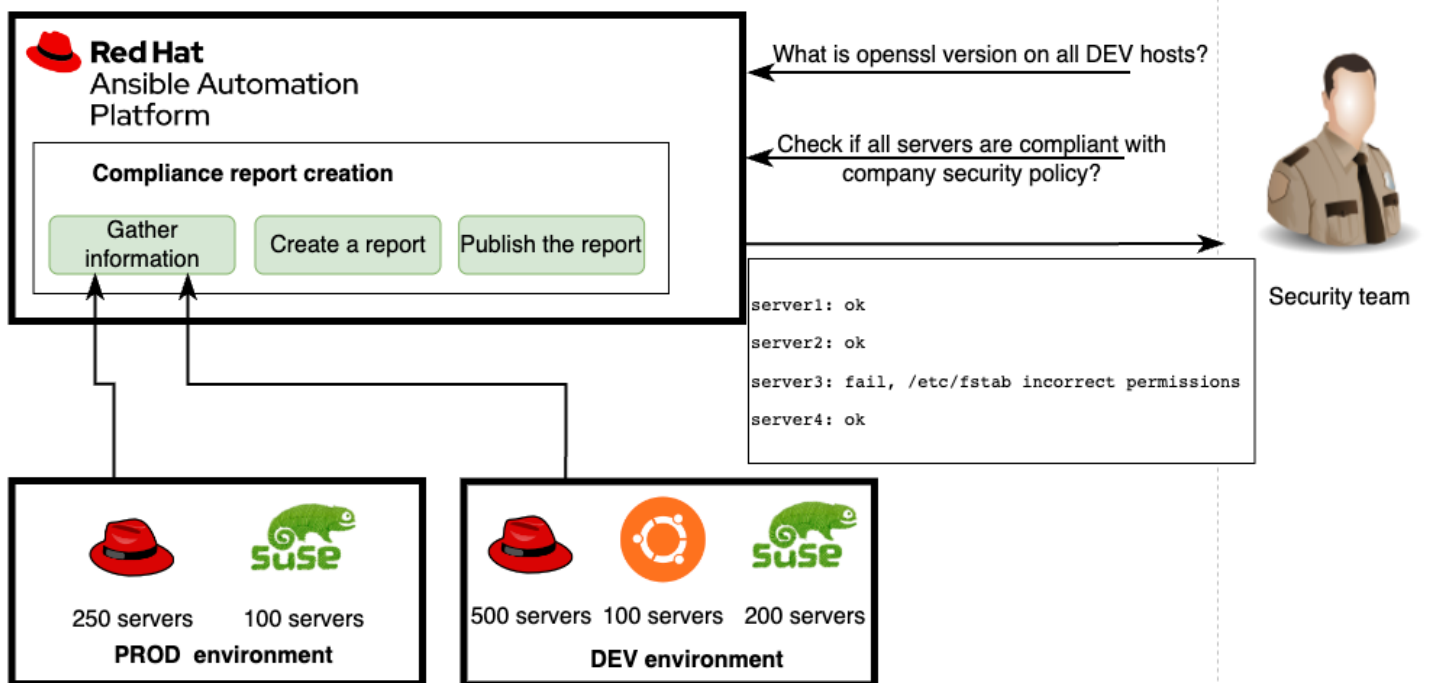
- Step 1 - Create a virtual server or public cloud instance
- Step 2 - Apply post configuration
- Step 3 - Deploy an application
- Step 4 - Update load balancer setting (if it applies)
- Step 5 - Create/update CMDB record



Patching and server upgrades



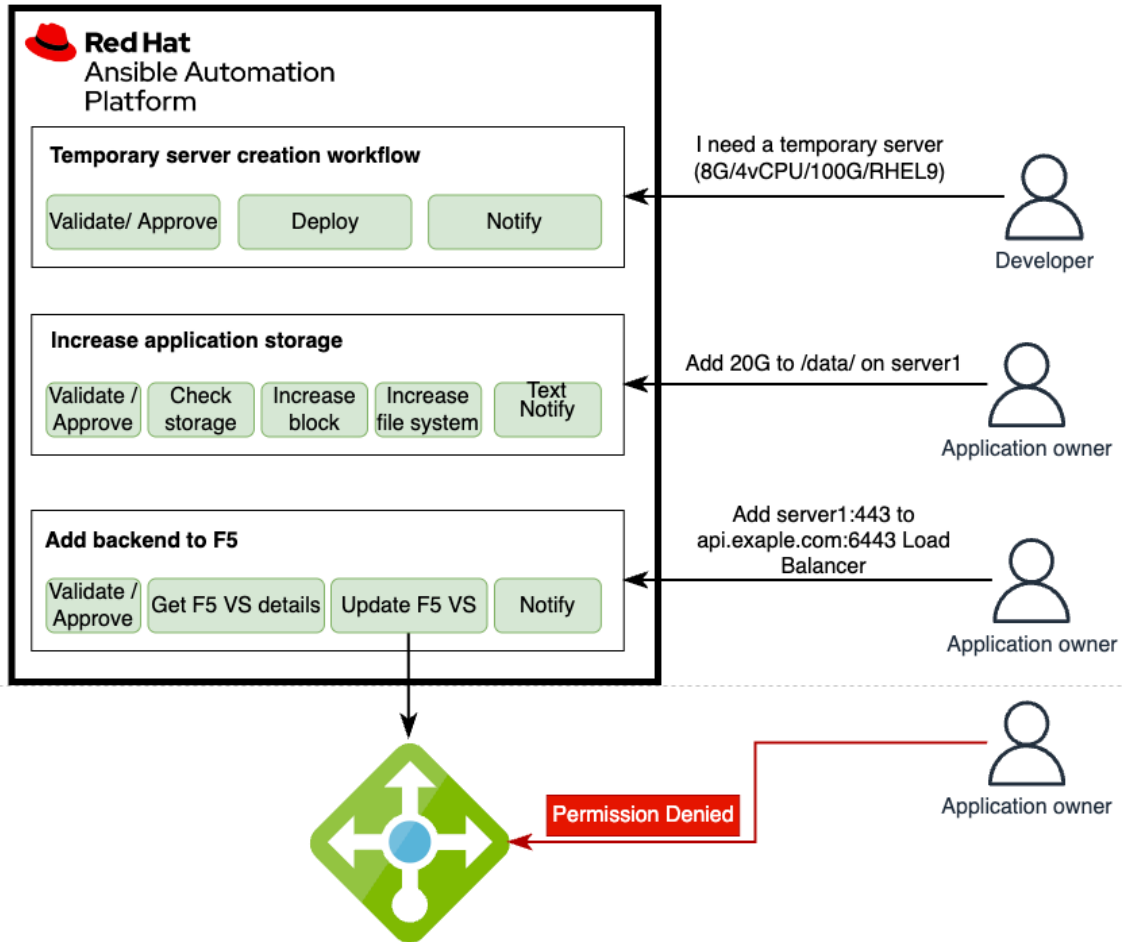
Security and Compliance



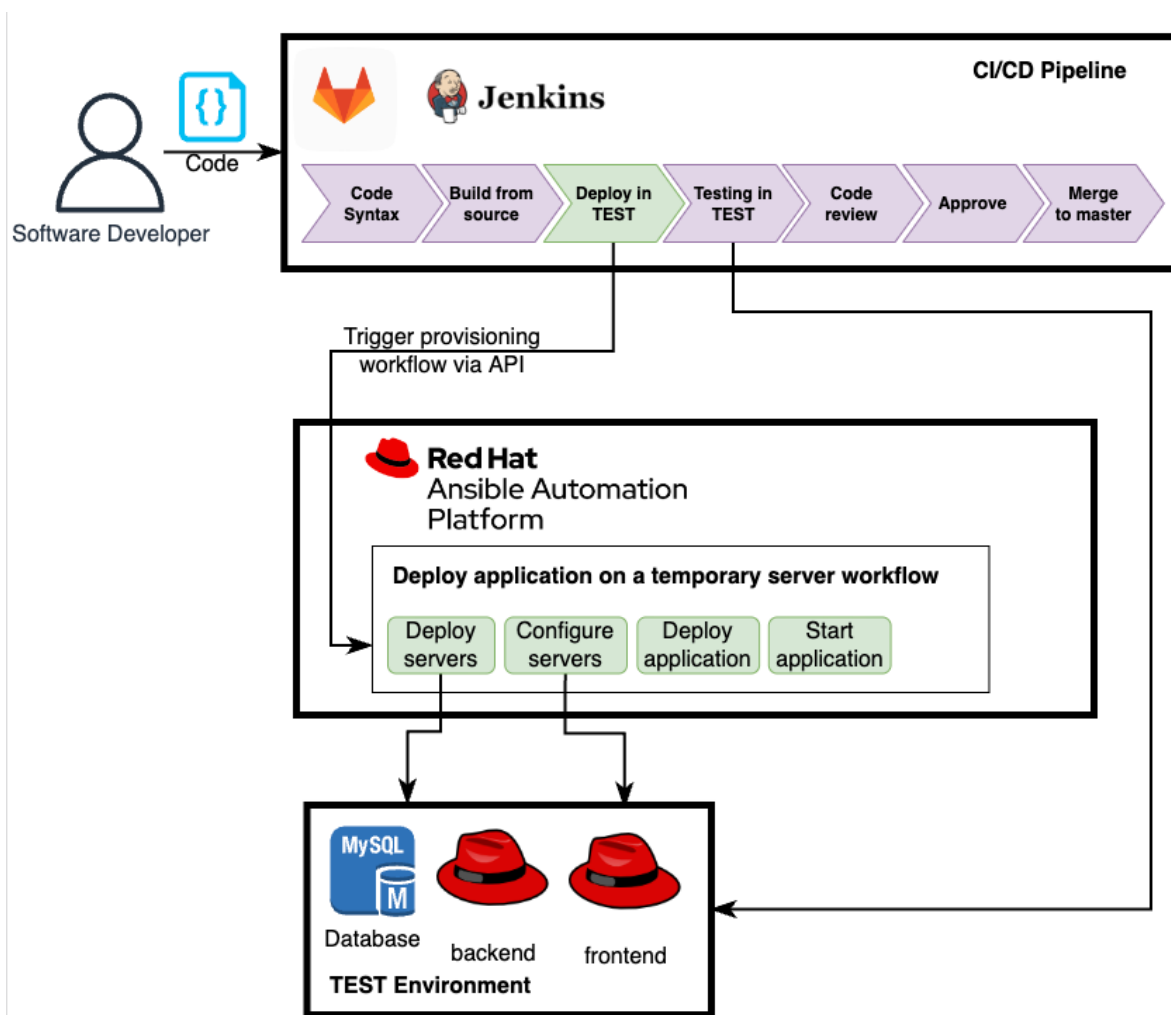
- Compliance reports from all hosts

- Configure security-related settings
- Enforce security baseline

User self-service



CI/CD pipeline integration



- Integrate Ansible Automation Job/Workflow templates with CI/CD workflows
 - Jenkins
 - GitLab CI
 - OpenShift pipelines (Tekton)

Deliverables

Ansible Automation Controller Single Node Deployment: Li9 team will install and configure Ansible Automation Controller on a standalone Virtual Server or Public Cloud instance.

Central Authentication using LDAP/AD: Li9 team will integrate Ansible Automation Controller with your existing LDAP or Active Directory service. Users will use their existing Active Directory (or LDAP) accounts to access Ansible Automation Controller.

RBAC using LDAP/AD: Li9 team will configure Role Based Access Control as follows:

AD or LDAP Group	Permissions
Automation Administrators	Members of this group will get Ansible Automation Controller Admin level access rights.
Automation Engineering	Members of this group will get admin access rights on the company's Organization
Automation Consumers	Members of this group will be able to use automation (like trigger a job, see the job results). However, they will not be able to change any Automation Controller objects (inventories, credentials, job and workflow templates, etc).

Note! We can use any AD/LDAP group names

Repository File Structure: Li9 team will create automation repository file structure that is aligned with Ansible best practices.

Ansible Playbooks/Roles to Automate Routine Tasks: Li9 team will develop a number of ansible playbooks(or roles) to automate routine tasks according to the use case. These automation playbooks will be pushed to a customer git repository to be consumed later from Ansible Automation Controller.

Ansible Automation Controller Configuration: The team will configure controller to allow automation consumers USE the automation. The following is included:

Type	Description
Inventories	List of hosts/devices managed by Ansible Automation
Projects	This entry gives access to automation playbooks

Credentials	Credentials to access managed hosts. Many credential types are supported by Ansible Automation. This can be user/password, or SSH private key depending on your infrastructure.
Job and workflow templates	This allows applying automation for hosts defined in an inventory using credentials and using automation playbooks from a project.

Customer responsibilities

The following items must be in place or provided for the Li9 team to quickly and successfully deliver the solution:

Share Manual Configuration Details for the Tasks: We expect that you will share with us all configuration details you want to automate. For example, if you need to configure syslog settings on all Cisco/Juniper/Arista network devices, we will expect Cisco/Juniper/Arista configuration details.

Access to Compute Platform or Servers: The project team needs temporary remote access to your compute platform to complete the work involved. Target platforms may vary - private or public cloud, on-premises servers or virtual machines.

Provide Service Prerequisites: The service requires a number of customer-provided items. Please review the full list of prerequisites in the “Service Prerequisites” chapter.

Assistance During Implementation: For the Li9 team to complete implementation work and debug issues we may uncover, we must have access to customer staff with responsibility for various infrastructure roles, including admin access rights to virtualization platforms, networking equipment, etc.

Timely Participation: To keep projects on schedule and ensure the necessary personnel will be available, both parties must adhere to agreed-upon plans and provide ample notice of any changes to those schedules.

Service Prerequisites and Dependencies

Red Hat Ansible Automation Platform Subscription: For the Pilot Project, you will need at least a minimal Red Hat Ansible Automation Platform 100 Node subscription, purchased from Li9.

Compute Platform Access: The solution assumes to use some compute and storage resources to deploy Red Hat Ansible Automation Platform, we will need the administrative access to perform the initial Red Hat Ansible Automation Platform deployment.

Version Control System: Ansible Automation Platform relies on Version Control. We would expect a GIT repository to be accessible with RW access rights. It would be great if you already have either GitLab/GitHub/BitBucket access or something similar.

Assumptions

- Li9 is on the customer approved vendor list
- All services are delivered remotely
- Manual configuration details are shared prior automation development
- All prerequisites are delivered on time before the start of the project. Missing or incomplete items may cause project delays
- Customer provides all necessary contacts, engineering support, and compute/network/storage resources continuously for a one-month time frame

Out of scope

- Ansible Automation Platform High Availability implementation
- Any compute, storage, or networking hardware deployments and configuration
- Any not related to deployment Virtualization platform configuration
- Any not related to deployment Public Cloud services configuration
- Any on-premises services configuration
- Any cloud providers infrastructure configuration, except needed
- Anything not listed in this document

