Red Hat
Enterprise Linux

Ansible Automation Mesh

Scaling Automation for the Enterprise

Brad Krumme
Senior Specialist Solution Architect

Red Hat

Agenda:

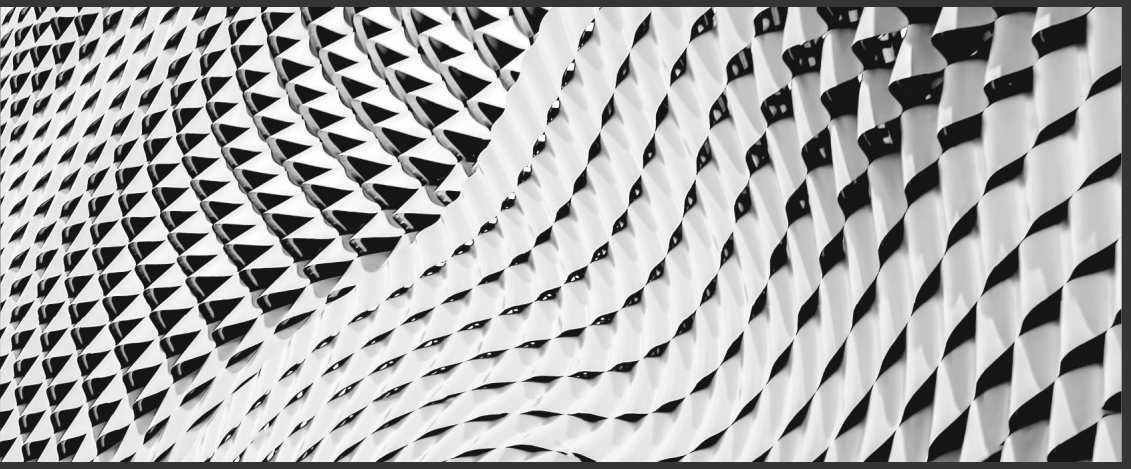
The Challenges of Scaling Automation

Automation Isolation: Execution Environments

Automation Mesh: Solving Enterprise Scalability

Demo

The Challenges of Scaling Automation



Red Hat

Enterprises are scaling globally, automation needs to scale in lock-step

Red Hat

Current execution challenges

Tightly coupled control and execution

Static capacity management

Control and execution planes scaled as a whole unit

Changing capacity requires downtime (outside of Openshift)

Control and execution cluster capacity shared

Managing execution capacity increases in complexity as it grows

Deployment limitations

Execution capacity limited to a single cluster

Closely coupled to the database and requires low latency

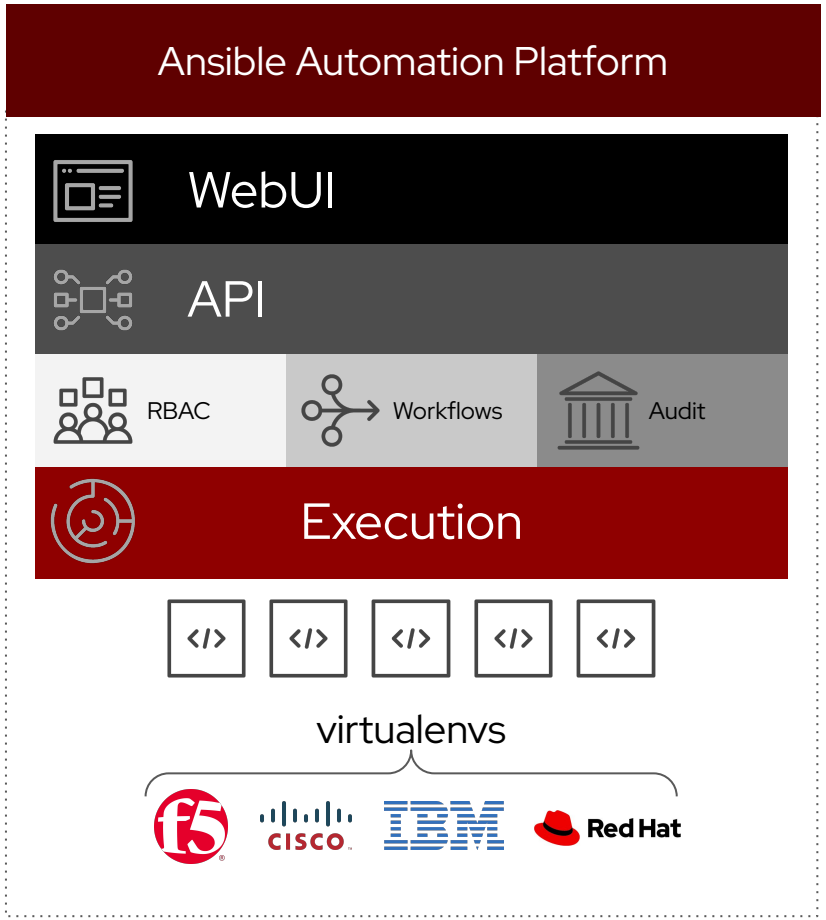
Isolated node constraints

One way communication from the controller cluster using SSH

Additional hosts needed for remote execution (Jump hosts, SSH proxies)

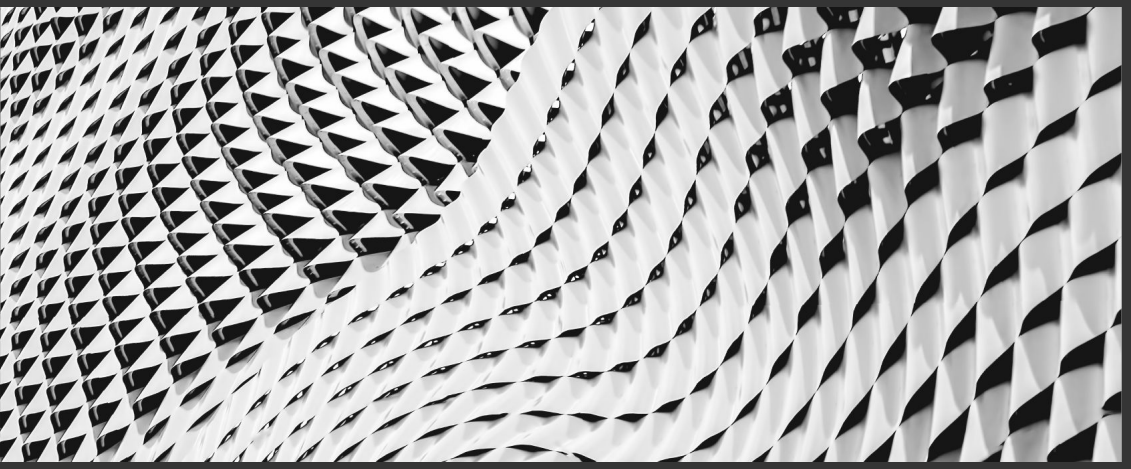
Extremely sensitive to latency

Low resilience to connection disruption



Red Hat

Execution Environments Revisited

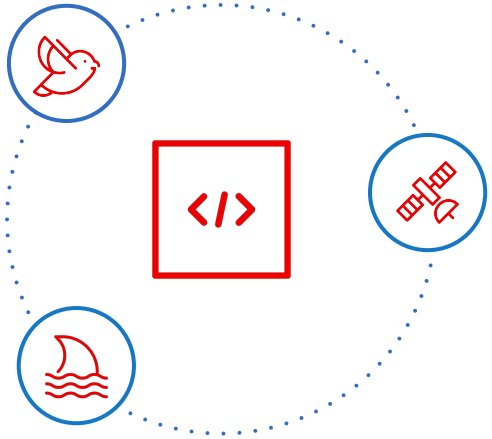


Red Hat

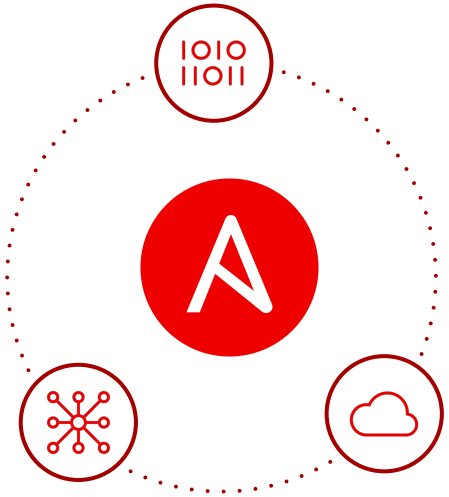
Many technologies, different life cycles

How to keep runtime environment, collections and dependencies aligned?

Collections



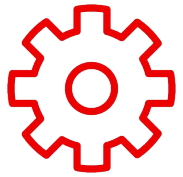
Dependencies



Runtime

Why Execution Environments?

Issues with Python virtual environments



Development Environment

Virtual environments (venvs) are not easily portable. The venv on your system likely does not match the production system.

Security and Standardization

Venvs are not secure. Some commands run inside the environment, some outside.

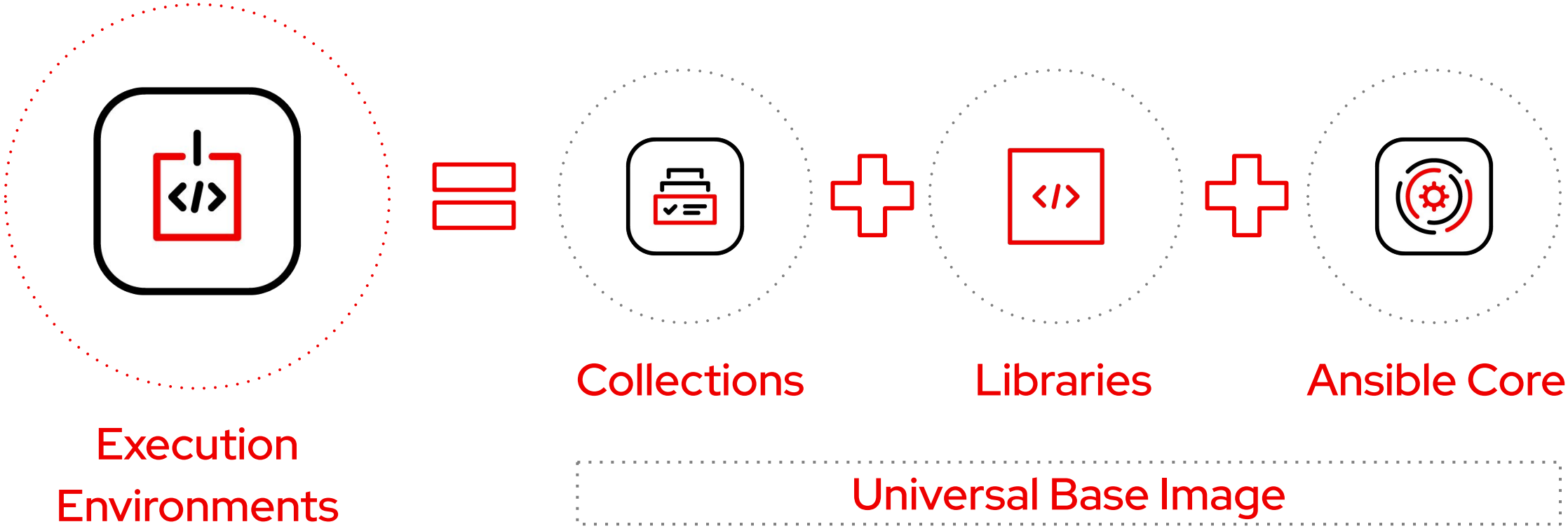
Venvs have to be moved between each Tower node in a cluster manually and whenever there's a change.

Maintenance

Collections and modules available to your venv may not be available in production. Maintenance of venvs is difficult and not documented well.

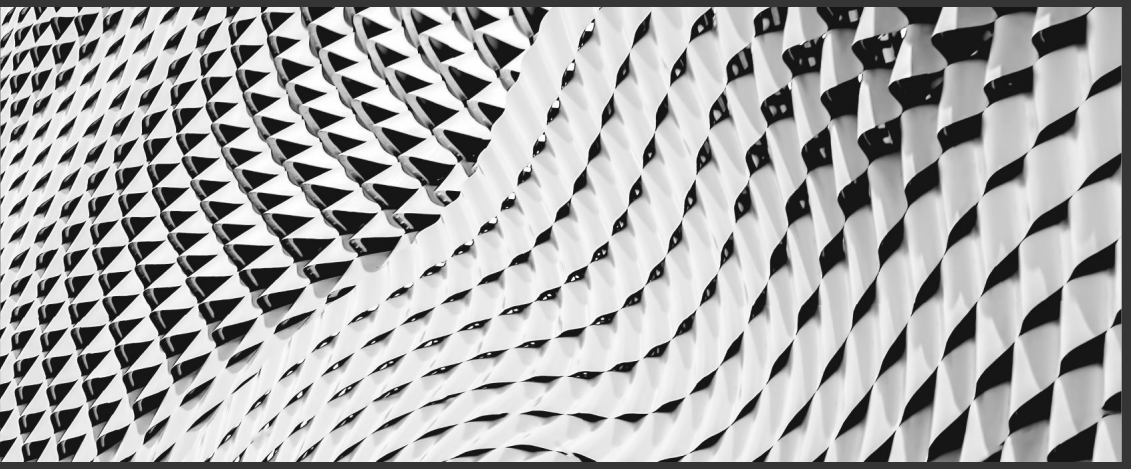
Automation Execution Environments

Components needed for automation, packaged in a cloud-native way



Red Hat

Automation Mesh:
Solving Enterprise Scalability



Red Hat

Automation mesh and automation controller

Simple, flexible and reliable execution scaling

Dynamic cluster capacity

Cluster capacity scales independently

Aims to scale and manage execution capacity without downtime

Decoupled execution and control plane

Deploy execution capacity where it's needed

Reduced operational overhead and control footprint

Scale globally

Execution plane resilient to latency and connection interruptions

Scale across segmented and remote networks

Natively build redundant mesh topologies

Improved communications and security

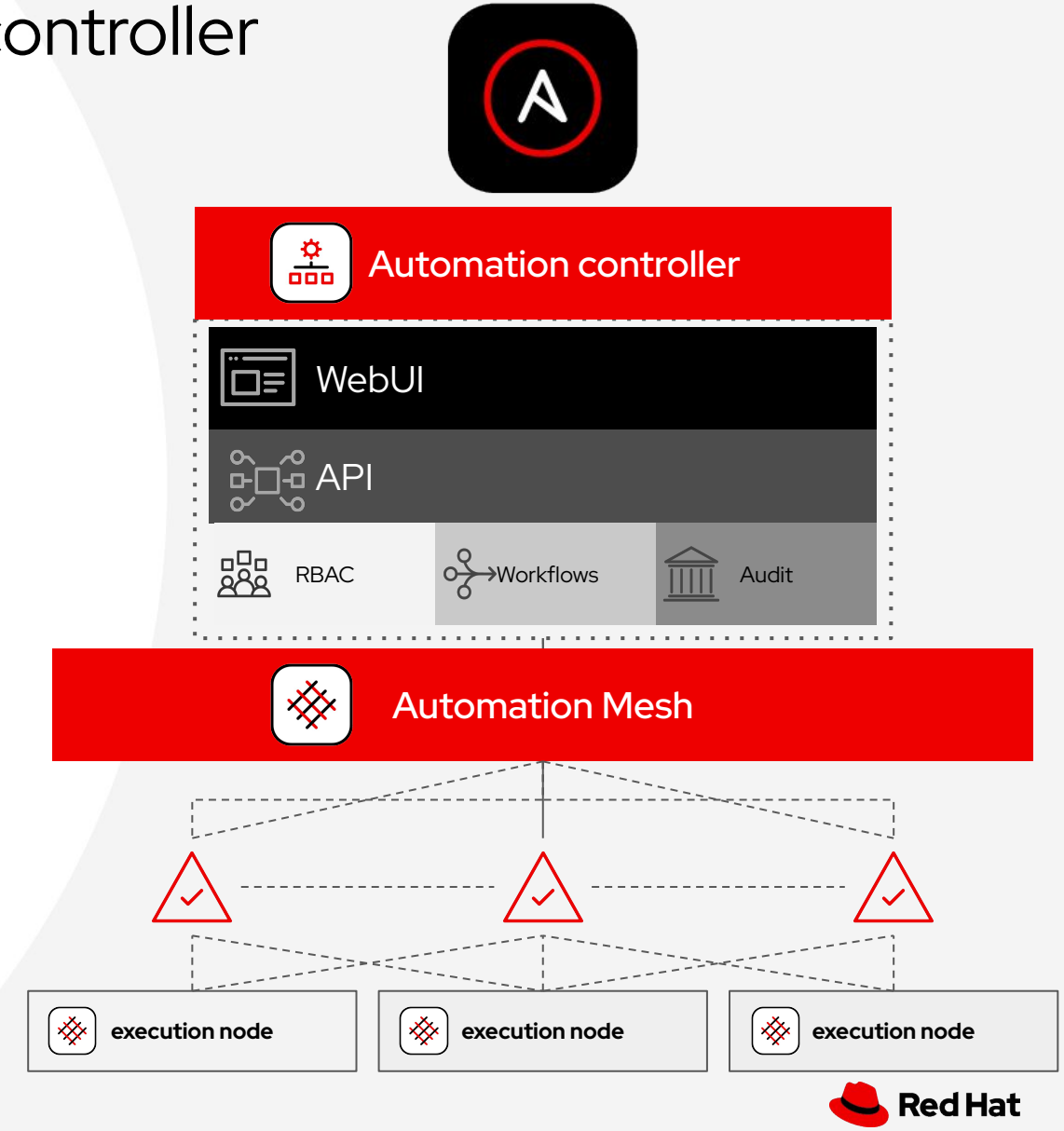
Bi-directional communication between execution nodes

Multi-hopped mesh communication capabilities

ACLs and digital signing of content

TLS authentication and encryption

Centralized management with automation controller



Automation mesh node types

Control plane

Hybrid

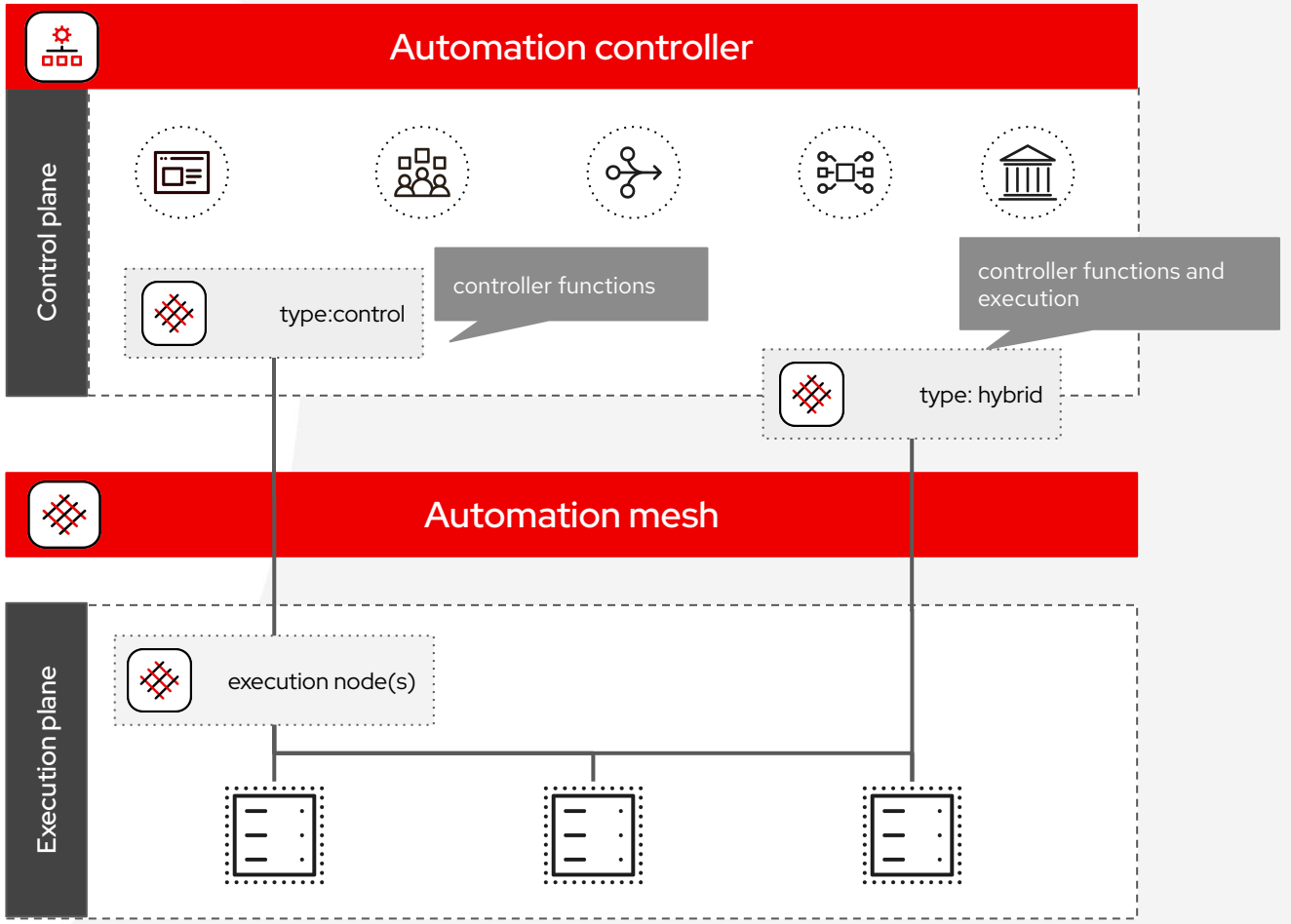
Default node type for controller nodes

Perform controller functions and execute automation

Control

Capacity is dedicated to controller functions

Automation execution capability is disabled



Red Hat

Automation mesh node types

Execution plane

What is the execution plane and why is it important?

Execution and hop nodes are collectively known as the execution plane

Run automation closer to the devices and systems that need it

Flexible designs possible across geographies and networks

Resilient to high latency and connection disruptions

Run automation without direct connection to controller

Execution node

Replaces Isolated Nodes and fulfills same functions

Dedicated to run automation on behalf of controller

No controller runtime functions executed

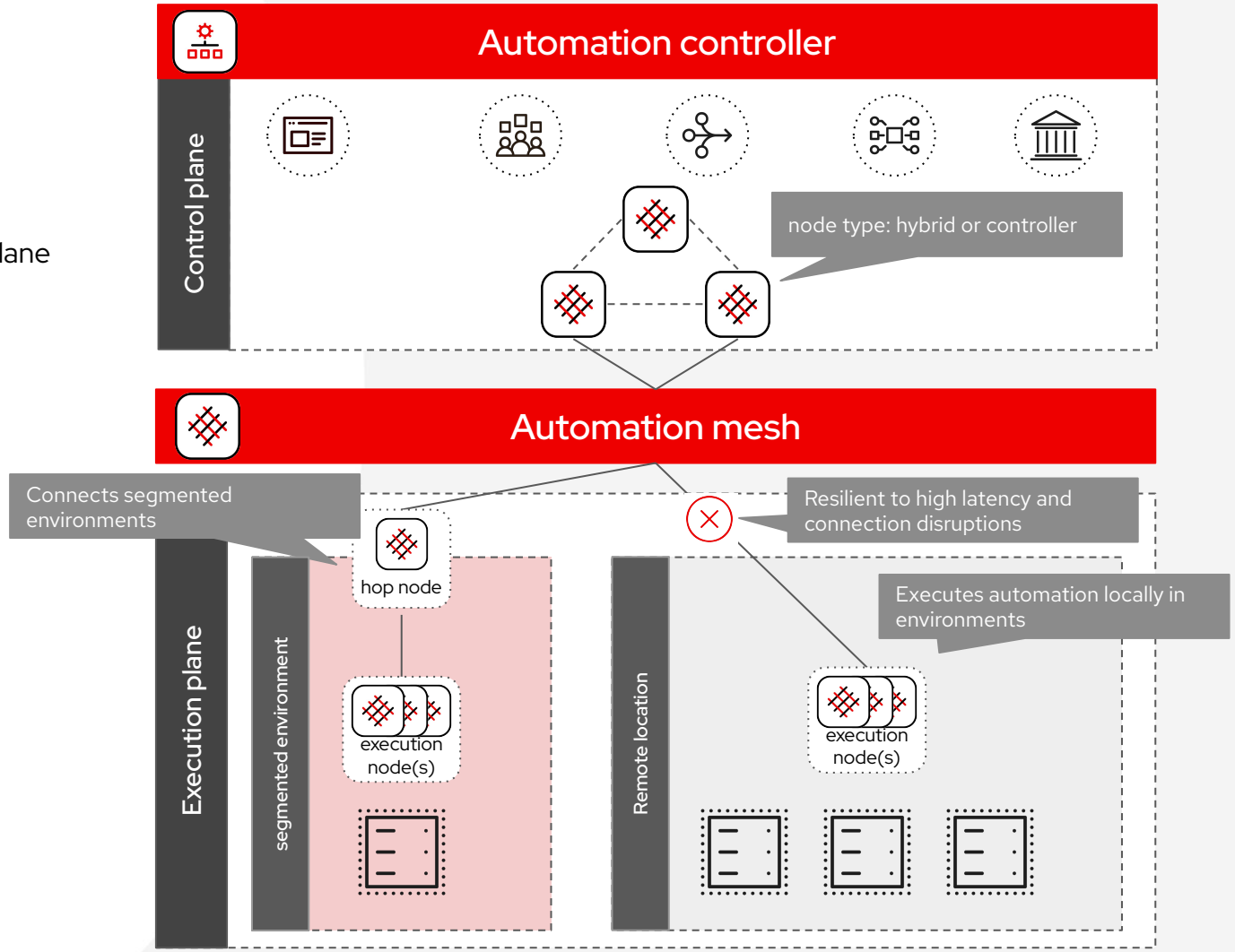
Job isolation via podman and execution environments

Hop node

Replaces need for jump hosts

Dedicated to route traffic to other execution nodes

Cannot execute automation

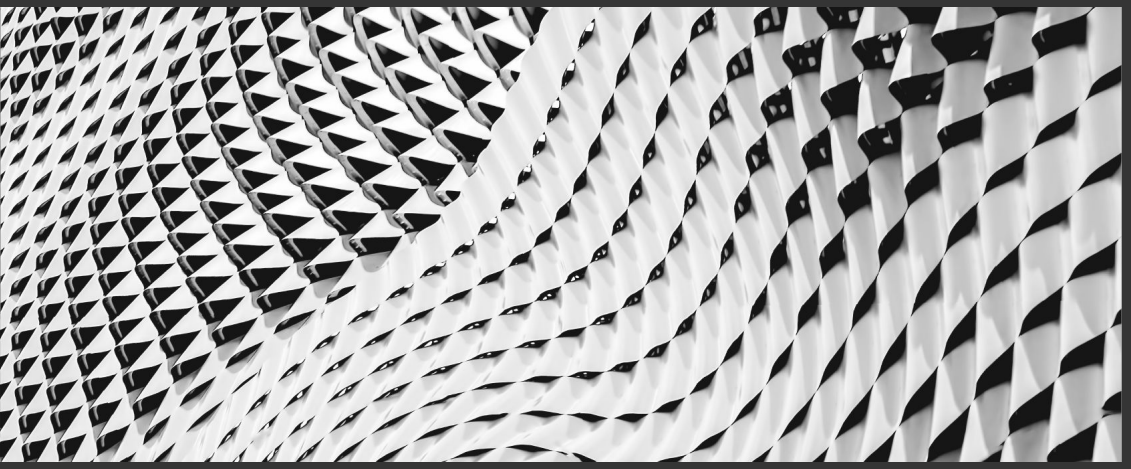


Red Hat

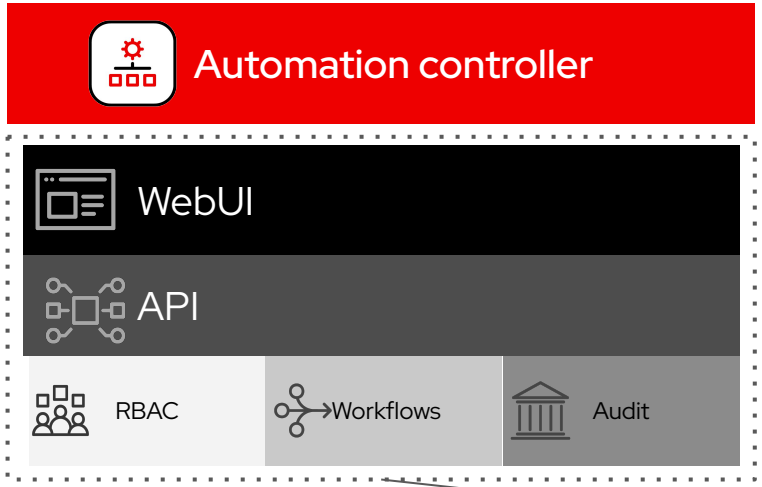
Enterprises are scaling globally, automation needs to scale in lock-step

Red Hat

Demo



Red Hat



Automation Controller On-Prem

Single Hybrid node with an external managed database

External DB so capacity can be added on-prem when needed

All API/UI/RBAC/Audit/Logging is done here

Managed Nodes

Red Hat Satellite is used as the inventory source

Systems are registered to Red Hat IDM

Low-latency network connection from Automation Controller

Automation Mesh Nodes in Amazon EC2

One Hop node to act as a gateway to the execution nodes

One Execution node to run automation jobs

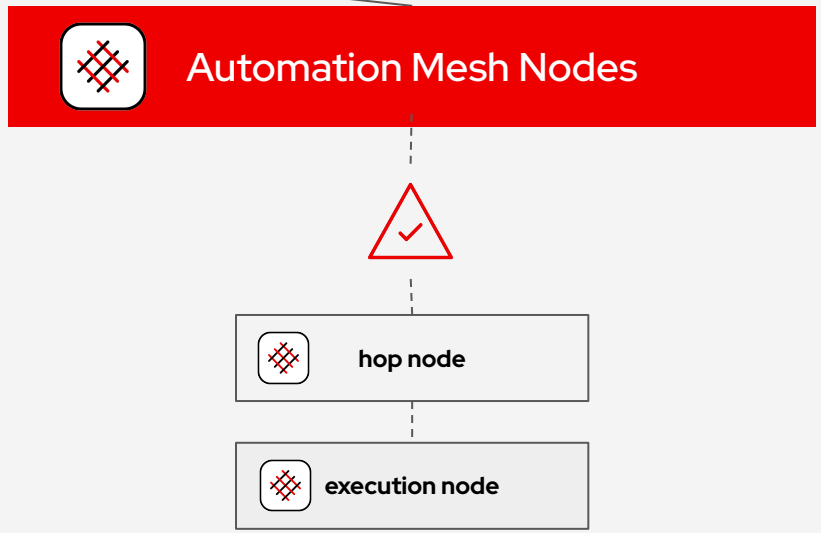
In a security group to allow SSH and Receptor traffic from Automation Controller

Managed Nodes

In a security group which blocks all inbound traffic *except* from the mesh nodes

EC2 is used as the inventory source

High-latency network connection from Automation Controller



Red Hat

```
[automationcontroller]
aap.bk.lab ansible_connection=local

[automationcontroller:vars]
peers=aws_hop

[execution_nodes:children]
aws_execution_nodes

[aws_execution_nodes]
aap-ue2-hop ansible_host=18.224.200.224 node_type=hop peers=aap-ue2-exec
aap-ue2-exec ansible_host=18.188.7.139 node_type=execution

[aws_hop]
aap-ue2-hop ansible_host=18.224.200.224

[aws_hop:vars]
peers=automationcontroller

[instance_group_local:children]
automationcontroller

[instance_group_aws]
aap-ue2-exec ansible_host=18.188.7.139

[database]
aap-db.bk.lab

[all:vars]
ansible_user=ansible
ansible_ssh_private_key_file="/root/.ssh/ansible"
admin_password='***********'

pg_host='aap-db.bk.lab'
pg_port='5432'

pg_database='awx'
pg_username='awx'
pg_password='*********'
pg_sslmode='prefer'

registry_url='registry.redhat.io'
registry_username='****************'
registry_password='****************'

receptor_listener_port=27199
```

Automation Mesh AAP Inventory

- automationcontroller group is for local hybrid or control node(s)
- execution_nodes group is for hop nodes and execution nodes either locally or in a separate network
- “Peers” variable configures which nodes communicate with each other
- instance_group_* groups create instance groups in Automation Controller
- **registry_username** and **registry_password** are *REQUIRED* to configure AAP to download Execution Environments
- **receptor_listener_port** is where the mesh receptor listens on all mesh nodes

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat