Multi-region Reference Architectures

Puppet Enterprise 2019.8

Introduction **1**

Multi-region reference architectures **2**
- Centralized deployment 2
  - In-region proxies variation 3
  - Distributed compilers variation 4
- Federated deployment 5
  - Trust relationships 5

Service delivery spanning network segments **6**
- Definitions 7
- Delivering services across segments 7
  - Other managed endpoints 8
  - Replica services 8
  - Compiler performance 8
    - Network Latency 8
    - Bandwidth 8

Use cases **9**
- Single pane of glass 9
- Autonomous data centers 9
- Cross-regional failover 9
- Low Bandwidth or Unreliable WAN 10

Exceptions to the reference architectures **11**

Document version: 1.1
Release date: 2021-05-11

Introduction

This document provides guidance for deploying Puppet Enterprise in multi-region or multiple network segment scenarios.

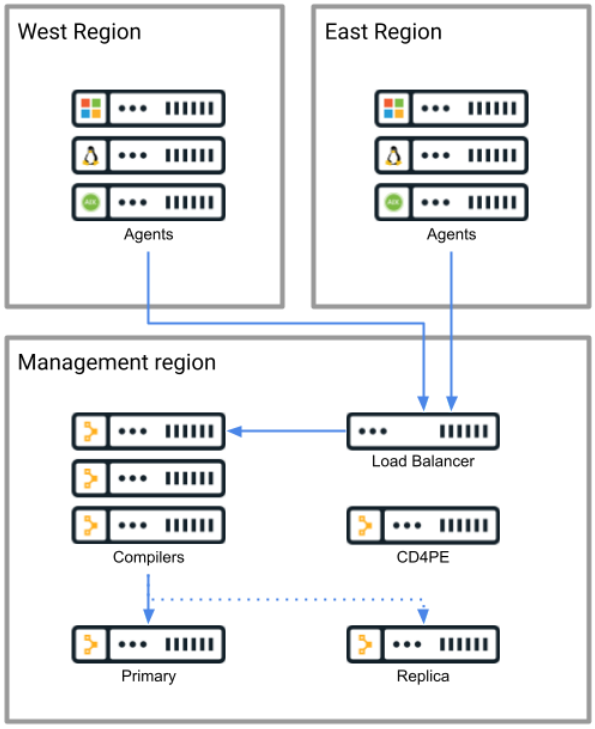
- The "Multi-region reference architectures" section describes two architecture patterns (and variants) for PE deployments spanning regions.
- The "Service delivery spanning network segments" section describes reasoning and generalized considerations for designing PE deployments spanning regions.
- The "Use cases" section presents a series of frequently encountered example use cases and how to apply the multi-region patterns to them.
- The final section provides some notes relating to exceptions to the presented reference architectures.

Multi-region reference architectures

Centralized deployment

A centralized deployment optimizes for single-pane-of-glass and ease of management across all regions, at the potential expense of availability or performance. Full Puppet services for a managed endpoint will require a connection between that endpoint and a Puppet compiler, and between that compiler and a Puppet primary server.

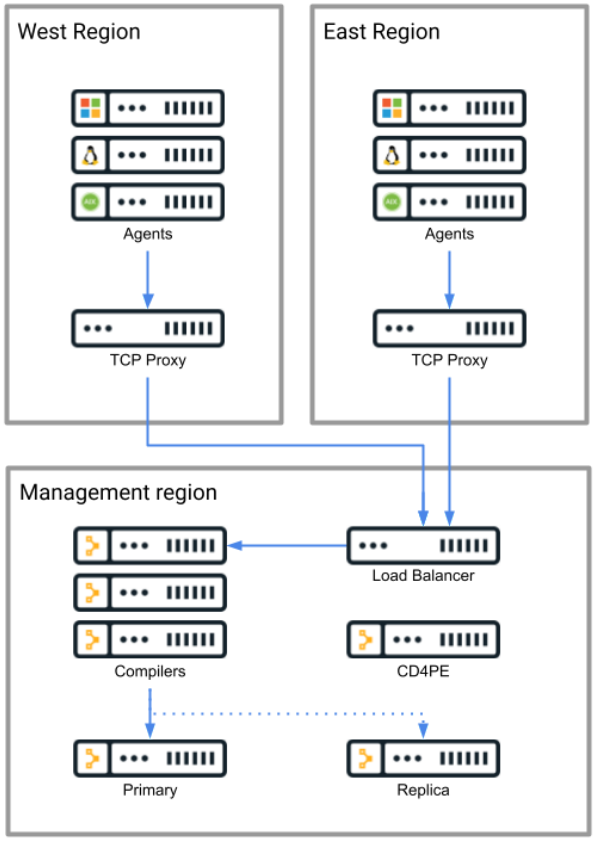
Significant latency or bandwidth restrictions between the compiler(s) and primary server may result in reduced overall performance for the centralized Puppet cluster.



In the most straightforward unified deployment, all compilers are deployed to the same region as primary and replica services. Agents from all regions connect to this centrally located service infrastructure for configuration management.

In-region proxies variation

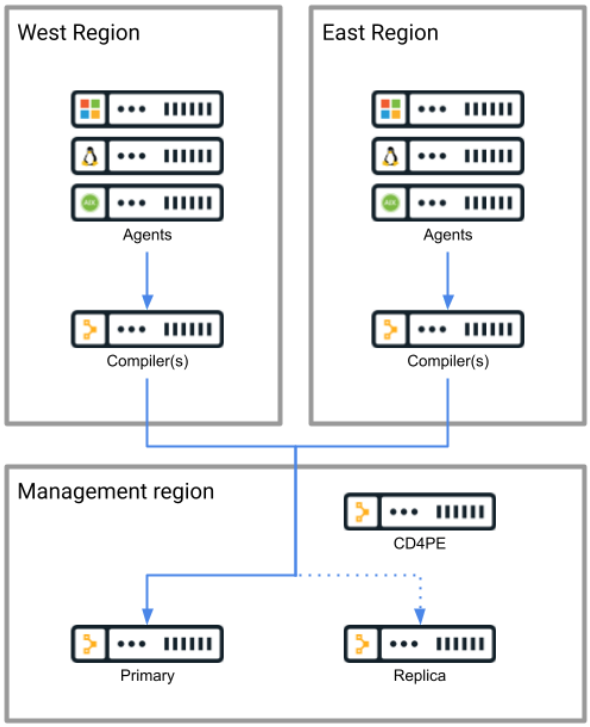
When network routing policies make it difficult for agents to connect directly to central infrastructure, consider using an in-region TCP proxy server. Network and firewall rules can be configured to allow the TCP proxy or proxies access to Puppet services, and agents can connect through the proxies. This permits the use of centralized compilers without requiring broad firewall openings in restrictive network situations.



SSL should not be terminated by these proxies. HAProxy is a common technology choice for this use case.

Distributed compilers variation

As a fallback alternative to direct connections or in-region proxies, compilers may be distributed and located across multiple regions, if serving regions which are linked with highly available, low-latency network connections.



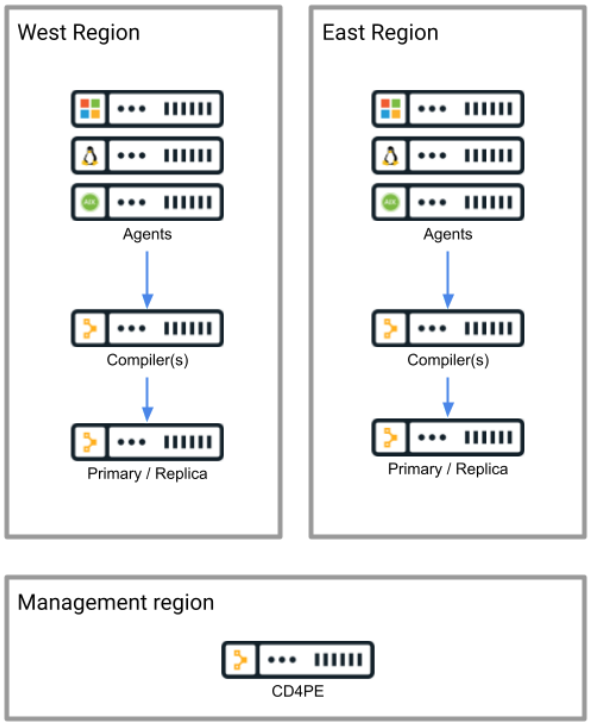
Note that this configuration comes with a caveat. Field experience indicates generally that low latency and reliable bandwidth between primary servers and compilers is necessary for optimal performance, and that high latency or bandwidth restrictions cause problems, but there is no data available defining the specific network conditions under which performance begins to suffer.

It is incumbent on customers when choosing this model to validate that Puppet platform performance with distributed compilers is acceptable for their use case.

Federated deployment

In a federated deployment, an independent Puppet cluster is deployed per-region. Regional availability and performance are maximized at the expense of losing a complete single-pane-of-glass Puppet management console for all regions.

A centralized CD4PE service is used to manage change deployments spanning all regions. This CD4PE service also provides limited federated reporting of node information from managed systems across all regions.



Optional integration with global reporting tools, such as Splunk or the Elastic stack, can be used to provide single-pane-of-glass visibility across all federated clusters.

Trust relationships

Normally, in a federated deployment, agents cannot automatically fail over to compilers or primaries in different regions. Agents should be “sticky”, and belong to one PE cluster only.

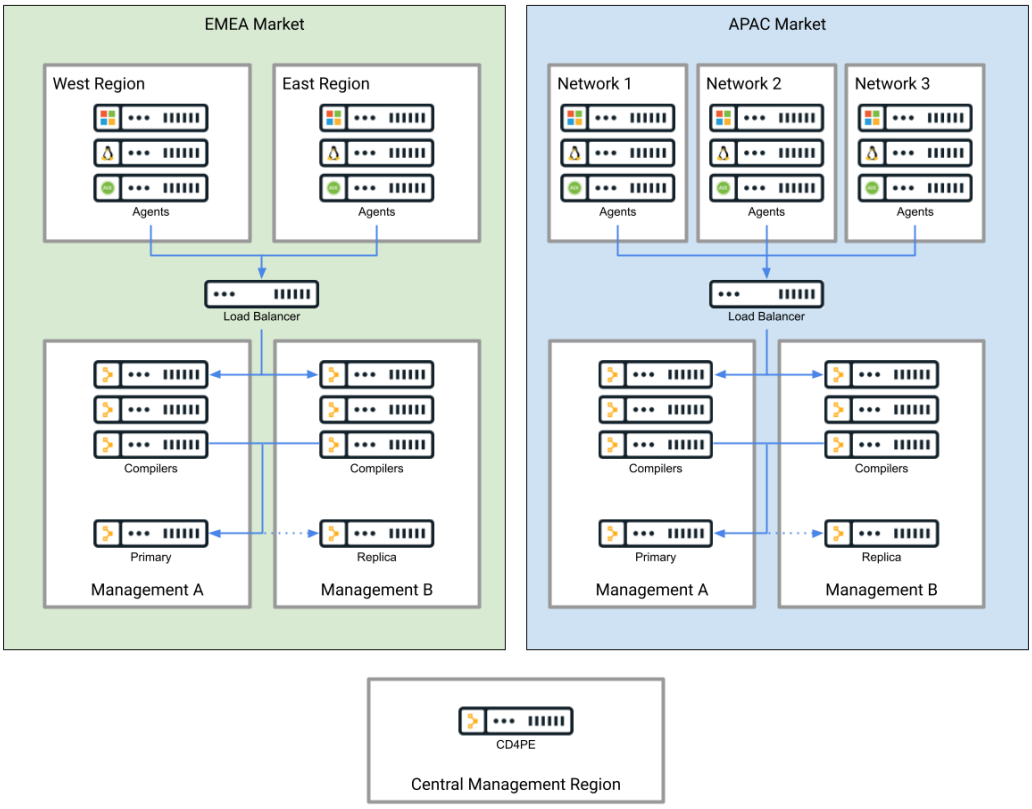
In the event agent fail-over is required, additional considerations must be made around duplication of node information in different clusters, and clusters must be specially configured to allow cross-cluster PKI trust. Customers needing to address these use cases are encouraged to reach out to Puppet Professional Services for assistance.

Service delivery spanning network segments

Designing a deployment architecture to deliver Puppet services spanning multiple network segments, such as availability zones, VPCs, DMZ(s), or network segments kept separate for any other reason, is informed by the available multi-region reference architectures defined above.

Holistic deployment designs spanning complex network segments, data centers, or global regions may be built from compositions of one or more of the reference models above. This section describes things to consider when creating or choosing compositions.

The following diagram illustrates an example deployment design incorporating both federated and centralized deployment models.



Definitions

Puppet Enterprise (PE) consists of primary services, optional compiler services, and optional replica services.

- **Primary services** must be deployed to a single network segment
- **Compiler services** may be deployed to multiple network segments
- **Replica services** may be deployed to a single network segment, which doesn't need to be the same segment primary services are deployed to

Puppet services must all be able to connect to each other according to the component and port requirements documented here.

Puppet services manage Puppet agents running on client servers, and also other managed endpoints addressable via ssh, winrm, or API.

- **Agents** must be able to reach compiler services
- **Other managed endpoints** must be reachable from primary services

Reference architectures for deploying PE are documented here.

Software architecture, outlining individual service relationships, is documented here.

Delivering services across segments

There are four ways to deliver services to agents located in separate network segments.

Option 1: Centralized deployment. Deploy compilers to the same network segment(s) as primary and replica services. Configure network routing to allow agents in other segments to connect directly to compilers through a TCP load balancer.

This is usually the best option.

Option 2: Centralized deployment, in-region proxies variation. This option may be a good one when a centralized deployment is desirable, but when it is easier to configure the required routing for a small number of in-region proxy endpoints, rather than configuring a route that all agents can use.

Option 3: Federated deployment. Deploy multiple separate PE clusters. For some use cases, fully independent clusters may be required to provide mandated levels of regional autonomy or cross-segment performance.

Federated deployment introduces some additional management overhead compared to centralized deployment options.

Option 4: Centralized deployment, distributed compilers variation. Deploy compilers to each separate network segment, for agents in those segments to connect to. Configure network routing to allow compilers to connect directly to primary and replica services.

See the compiler performance section below for requirements to consider before deploying compilers to other segments.

Other managed endpoints

All outgoing connections to non-agent managed endpoints (for example, network or storage devices managed through an API) originate from the primary Puppet server. This means that to manage an ssh, winrm, or API endpoint, that endpoint must be reachable from the primary and replica servers that manage it.

Replica services

A single DR replica can be deployed per PE cluster. The replica may be deployed in a region or network segment different from the primary, provided that network connectivity between regions is reliable enough to support streaming replication data.

When both a replica and compilers are present in an alternate region, this can provide partial service availability in that region, in the event connectivity to the primary region is interrupted. Full service will be restored either when connectivity to the primary region is restored, or if the replica is promoted to primary.

Compiler performance

Puppet compilers run a PuppetDB service, which uses the PostgreSQL frontend/backend protocol to communicate with a PostgreSQL server, which in turn is part of primary services. This protocol is heavier and passes more traffic than the Puppet agent-to-server REST API calls which it supports. For this reason, agent-to-server traffic is more forgiving of network errors and slowdowns than compiler-to-primary-services is.

It is recommended that customers choose deployment architectures which ensure low-latency, reliable bandwidth networking between primary services and compiler services.

Network Latency

High network latency between primary services and compilers may introduce performance problems that can reduce performance for affected compilers, and in some circumstances even have negative effects on performance across the cluster. Compilers should ideally not be located in network segments with high latency back to the primary services.

Bandwidth

Generally speaking, bandwidth requirements are calculated based on the full path from agents to primary services, regardless of where compilers are located. There may be minor cross-segment bandwidth savings by locating compilers in the same segments as the agents they serve, based on Puppet fileserving, but for most users the difference will be inconsequential.

Use cases

The following section describes a series of common scenarios, requirements, and which multi-region architecture is appropriate to meet those requirements.

Single pane of glass

In a single pane of glass use case, customers would like the ability to view or control their Puppet service from a management console or endpoint that presents data from all managed endpoints in a unified interface.

To achieve a complete pane of glass, use a centralized deployment (either variant).

Autonomous data centers

Some customers have a requirement that each of their data centers (or a subset of their data centers) should be fully autonomous. That is, they should never depend on non-local services, or services located in a different datacenter. WAN link failures should have virtually no impact on a data center's normal operational status.

On occasion, this ask may be related to permissions more than resilience. This may be due to something such as regulatory laws of a specific country in which the customer has infrastructure being managed.

To achieve fully autonomous PE services in different data centers, use a federated deployment.

Cross-regional failover

When customers have the requirement for cross-region failover and aPuppet primary services outage occurs in one data center, Puppet agents and other client services should failover to Puppet primary services located in another region, or data center. Depending on which other use cases this use case is combined with, cross-regional failover aims to achieve decentralization of Puppet primary services, without strict federation.

To achieve cross-regional failover for two regions only, with fast, reliable network connectivity between them, it may be possible to use a centralized deployment with distributed compilers and replica services deployed to the second region.

If more than two regions are required, or if network connectivity is unreliable, use a federated deployment, configured with mutual PKI trust relationship(s) between failover regions. Note that this deployment has a high operational complexity relative to other options. Supporting failover will also require careful auditing of how features like puppetdb

queries and exported resources are used in Puppet code, to prevent unexpected configuration changes from occurring when an agent fails over to a different Puppet deployment.

Low Bandwidth or Unreliable WAN

Some customers have infrastructure spread across regions where the connectivity between sites is unreliable or the available resources (bandwidth) contentious.

To provide services under these circumstances, a federated deployment is usually the correct option.

A centralized deployment may be used, if it is acceptable for connectivity to managed endpoints in sites besides the one where PE is deployed to be unreliable. Because Puppet provides eventual-consistency configuration management, this may be a reasonable option for some use cases.

Exceptions to the reference architectures

Uncommonly, situations may exist where customization is required to fit a Puppet Enterprise deployment to the constraints of a network situation. This may include actions such as configuring and deploying “remote compilers”, optimized for placement in network segments with high-latency connections, PuppetDB-optimized core compilers to balance them, and other such adjustments.

The multi-region reference architecture building blocks exist to simplify and standardize deployment of Puppet Enterprise. Significant deviation from these guidelines should only be performed under the advisement of Puppet Support or professional services.