MORPHEUS

Morpheus Data 5.x Reference Architecture

Version 3.1
Last Updated: January 2023

VERSION HISTORY

Description	Version	Date
Original Document	1.0	20 Nov 2019
Updates including AWS supported architecture	2.0	12 May 2020
Updated to include 4.2.2 architecture changes, added options to disaster recovery section, minor updates and restructured the document for easier readability.	2.1	6 Aug 2020
Updated to include architecture changes to 5.3.2, updated compatible operating systems, updated two-factor authentication support details	3.0	2 Aug 2021
Added navigable bookmarks. Updated formatting alignment and minor spelling and grammar. Added inter-document links. Updated document metadata. Updated *Deployment Requirements > Operating Systems* to align with public documentation. Added iconography for notes and warnings. Removed stale URL links. Removed messaging tier information for Morpheus Data Appliance versions 4.x. Removed reference to AWS multi-AZ architecture as a multi-site architecture. Added AWS Single Server (All-in-One) as a supported architecture. Updated AWS icons in reference architecture diagram. Updated *Morpheus on AWS* reference architecture diagrams and verbiage. Removed AWS service descriptors from *Morpheus on AWS* section. Removed references to Redis used in v4.x. Removed references to RDS Cross-Region Replication. Updated server sizing to remove references to 8GB-RAM systems.	3.1	05 Jan 2023

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

DEPLOYMENT REQUIREMENTS

Resources

Use the recommended resources details within the sections below to achieve acceptable performance for production environments and continue to monitor resource utilization on an ongoing basis increasing resources or scaling the deployment as necessary. Minimal resources are adequate for non-performance / non-production based testing.

Licenses

Community License (Morpheus Community Edition)

Available to all users installing Morpheus Data Appliance version 4.1.1 or later. A Community License is a 12-month time-limited license that restricts the user to three integrated clouds and up to 25 workloads managed or discovered. Guidance (rightsizing) recommendations are read-only. This license is fully self-service and does not include any Morpheus support services. Community licenses may be requested from the Dashboard tab of Morpheus Hub (morpheushub.com) so long as the account does not currently have other licenses assigned.

PoC License

Typically limited to 30 days to allow customers and Morpheus field teams to work together in testing suitability for specific automation use cases and environments.

Production License

Production licenses are good for the duration purchased and are based on *Workload Elements*. Workload Elements are defined as the granular unit of compute that is directly associated with an application service. Workload elements include both discovered and provisioned instances in any attached clouds. For a more thorough explanation of workload elements, review the knowledge base article at the following URL:

https://support.morpheusdata.com/s/article/What-is-a-Workload-Element-or-WE-for-purposes-of-Morpheus-licensing

Component Licenses

Morpheus Data provides support services for the Morpheus platform and limited support for the underlying open source components when deployed in compliance with the supported architectures defined herein.

Refer to the *Components > Open Source section of this document* for a full listing of all open source components.

If deploying a Multi-Site Active/Passive Architecture, Elasticsearch Cross Cluster Replication requires the purchase of an Elasticsearch Platinum Level subscription directly from Elasticsearch.

Repositories

Access to base “*yum*” and “*apt*” repositories (customer or publicly hosted) is required in order to deploy the Morpheus Data Appliance.

Operating Systems

The following operating systems are supported for the latest version of Morpheus.

Amazon Linux	v2
CentOS	7.x, 8.x (stream), 9.x (stream)
Debian	9, 10, 11
RHEL	7.x, 8.x, 9.x
Oracle Enterprise Linux (OEL)	7.x, 8.x
SUSE SLES	12, 15
Ubuntu	16.04, 18.04, 20.04, 22.04

Networking Requirements

Connectivity

Requirements for installation:

- Network connectivity from the administrator workstation(s) to the Morpheus node(s) over TCP port 22 and TCP 443.
- Network connectivity from the Morpheus node(s) to the yum/apt repos
- Network connectivity from the users to the appliance over TCP 443 (HTTPS).
- Virtual Machines and Docker-based hosts must be able to reach the Morpheus node(s) IP address on TCP 443

ℹ Note: Additional port and protocol requirements maybe found in *Appendix B of this document*.

Name Resolution

Morpheus node(s) must be self-resolvable to their own hostname, FQDN, and static IP address prior to installation. Managed machines must be able to resolve the Morpheus appliance.

Latency

In a distributed architecture latency under 5ms is strongly recommended for acceptable performance.

SECURITY

Authentication

Morpheus is capable of integrating with several single sign-on (SSO) solutions. These integrations require mapping security groups to user roles in Morpheus ensuring proper role assignment at first login. For details of the current compatible identity provider integrations, please see the following URL:

> https://docs.morpheusdata.com/en/latest/integration_guides/IdentityManagement/IdentityManagement.html

Morpheus is also capable of supporting multi-factor authentication (MFA) solutions. Local native Morpheus accounts can be configured for two-factor authentication (2FA). Accounts seeded through a SAML 2.0 integration which supports multi-factor authentication can also be used.

Encryption

Data in the Morpheus Data Appliance database is protected via AES-256 symmetric key encryption. Protected data includes configuration data, passwords, and configuration metadata. Data at rest on the file system is not client-side encrypted but can exist on an encrypted file system.

FIPS 140-2

Morpheus Data Appliance version 3.6.5 and later supports Federal Information Processing Standard (FIPS) 140-2. If FIPS compliance is required, customers should ensure they are using the Morpheus FIPS installer.

Certificates

The Morpheus Data Appliance supports using self-signed certificates or Certificate Authority (CA)-signed certificates. Self-signed certificates are ideal only for testing purposes. CA-signed certificates are preferred and recommended for all other purposes.

⚠ If integrating with Active Directory, a valid trusted CA-signed domain certificate is required.

Compliance

Each minor software version of the Morpheus Data Appliance (e.g. 5.*1*, 5.*2, etc*) is scanned for Common Vulnerabilities and Exposures (CVEs) and remediated prior to release. When necessary, patch versions (e.g. 5.1.*1*, 5.1.*2, etc*) are released to address immediate CVEs. CVEs addressed in each version are documented in the Morpheus Data Appliance *software release notes*.

Log Types

The Morpheus Data Appliance is capable of servicing large amounts of log traffic by utilizing Elasticsearch and buffered log transmission protocols. Morpheus provides a highly efficient and highly scalable solution for capturing log data. Logs can also be forwarded to external third-party log services.

Morpheus Server Log

Morpheus server logs are rotated every 24 hours with 30-day retention. These logs include *check server*, *guacd*, *elasticsearch*, *mysql*, *nginx*, *rabbitmq*, and *redis* (if applicable).

Audit Log

The audit log documents system changes made by users. For example, create and delete instances. Audit logs are stored on the file system with the same retention as the Morpheus Server Logs. The Activity feed, visible through the UI, is a subset of the Audit log which is stored indefinitely in the database.

Agent Log

Application logs are sent to the Morpheus Server from managed machines with the Morpheus Agent installed. ElasticSearch is used for this purpose. The Morpheus Agent forwards syslog messages to Morpheus with default retention of 7 days.

Telemetry

Morpheus Data has the ability to receive customer's telemetry data for analysis purposes. Transmission of telemetry data can be disabled as a license feature if desired.

Languages

The default language of the Morpheus Data Appliance web UI is English. Language settings can be changed on a per-user basis by editing the User Settings for the user.

COMPONENTS

Open Source Software in the Morpheus Data Appliance

Morpheus Data utilizes open-source components that are reviewed for updates at every major release (e.g. 5.*1*, 5.*2, etc*). In between these major releases, if a Common Vulnerability and Exposure (CVE) necessitates an update, it will be done in a minor release (e.g. 5.1.*1*, 5.1.*2, etc*).

Morpheus Data Appliance version 5.x uses open-source software listed below.

- Apache Guacamole
- Apache Tomcat
- Elasticsearch
- Erlang
- Java Open JDK JRE
- MySQL
- Nginx
- RabbitMQ
- Redis (Morpheus version 4.1 or prior)

More information regarding Morpheus Data's use of open-source software and licenses can be found on the Morpheus Data website at https://www.morpheusdata.com/licensing

⚠ Care should be taken to ensure component version compatibility is maintained during administrative operations including upgrades and updates. Compatible versions can be found in the release notes of each software release, under the Service Version Compatibility section. For an example, see the following URL:

https://docs.morpheusdata.com/en/latest/release_notes/compatibility.html#morphver-service-versions-compatibility

Morpheus Data Agent (Optional)

The Morpheus Data Agent is lightweight, secure, and is available for Linux-based and Windows-based workloads. The Agent provides greater visibility into logs and stats and is capable of processing instructions initiated by the Morpheus Data Appliance.

The Morpheus Data Agent initiates an outbound connection from the managed workload to the Morpheus Data Appliance over TCP port 443. This establishes a bi-directional command bus enabling orchestration of workloads without additional credentials or access protocols like SSH or WinRM. The Morpheus Agent may be installed over Cloud-init, Windows unattend.xml, VMware Tools, SSH, WinRM, Cloudbase-init, or manually.

ℹ The Morpheus Data Agent is not required by Morpheus to manage an instance. However, without the Morpheus Data Agent, the accuracy of statistics will vary based on the integrated cloud's capability. SSH or WinRM connectivity and credentials will be required for managed workloads without the Morpheus Data Agent installed.

Capabilities

The Morpheus Data Agent performs the following functions:

- Provides a command bus which allows Morpheus to orchestrate automation on managed machines without credentials
- Accepts and executes commands and scripts
- Provides connection persistence over HTTPS web socket and runs as a service
- Buffers and compresses logs and sends them in chunks to minimize packet transfers

Supported Operating Systems

For the most-recent list of operating systems supported by the Morpheus Data Agent, see the online Morpheus documentation at the following URL:

https://docs.morpheusdata.com/en/latest/getting_started/functionality/agent/osSupport.html

MORPHEUS DATA APPLIANCE SERVICE TIERS

The Morpheus Data Appliance includes four service tiers – the *Application Tier*, the *Messaging Tier*, the *Non-Transactional Database Tier*, and the *Transactional Database Tier*. Morpheus supports various architectures in which these tiers can run on a single machine, distributed across machines, or clustered for redundancy. An explanation of each tier follows.

Application Tier

The Application tier runs NGINX, Tomcat, and Guacamole services.

NGINX

The NGINX component of the Morpheus Data Appliance provides SSL termination and cache proxy for the Tomcat container. NGINX is open-source software built for web serving and is designed for maximum performance and stability. NGINX uses an asynchronous event-driven approach, rather than threads, to handle requests. NGINX's modular event-driven architecture provides for more predictable performance under high loads.

Apache Tomcat

The Apache Tomcat component of the Morpheus Data Appliance hosts the user-interactive application. Apache Tomcat is an open-source implementation of *Java Servlet*, *JavaServer Pages*, *Java Expression Language*, and *WebSocket* technologies.

Apache Guacamole

The Apache Guacamole component of the Morpheus Data Appliance provides a clientless remote console for instances, hosts, virtual machines, and bare metal machines. Platform type and cloud settings determine the port and protocol used for remote console connections. Guacamole is capable of providing a remote console via SSH, RDP, or VNC over associated standard ports.

High Availability (HA) for the Application Tier

To provide redundancy, Application Tier servers must be placed behind a load balancer. HA Application tier servers require shared storage at */var/opt/morpheus/mopreus-ui/** on each server. This shared storage contains uploaded virtual images, deployment archives, logos, Ansible, Terraform, and Morpheus backups.

The Application Tier can scale vertically or horizontally assuming all servers have access to the shared storage. The Application Tier runs only "stateless" services. All communications between tiers and workloads go through the Application Tier, apart from each tier's inter-cluster communications.

Shared Storage

Redundant configurations of the Application Tier require a shared file system so that all nodes within the Morpheus cluster are able to connect to necessary files such as white label images, uploaded virtual images, deploy uploads, Ansible plays, Terraform, and Morpheus backups. This storage can be externalized to an object storage service or a simple NFS cluster.

Load Balancer

Load balancing for the Application Tier over TCP port 443 is required for highly available solutions. Session persistence should be enabled. The *Least Connections* load-balancing algorithm is recommended for use with the Application Tier as it is typically used when session persistence is enabled since traffic can become unevenly distributed if other algorithms such as *Round Robin* are used. Health Checks should be enabled to redirect users to active servers. SSL pass-through is supported for use with Morpheus; however, SSL offloading is not.

Non-Transactional Database Tier

The Non-Transactional Database Tier consists of Elasticsearch. The Elasticsearch component of Morpheus enables logs and metrics from managed and discovered machines. Elasticsearch is a distributed, RESTful search and analytics engine that is capable of high write throughput at scale. If a distributed architecture is desired, an Elasticsearch cluster must be created.

High Availability (HA) for the Non-Transactional Database Tier

Minimally, a three-node cluster connected over transport is required for Elasticsearch high availability. Elasticsearch nodes can be added to the cluster to increase its capacity and reliability. To preserve performance, the nodes of an Elasticsearch cluster should be on the same IP network within the same datacenter. Elasticsearch should have multiple master nodes to avoid split-brain scenarios. Master nodes store detailed cluster state, while data nodes are responsible for storing and querying the actual index data. By default, an Elasticsearch node is both a data node and eligible to be elected as the master node that controls the cluster.

Elasticsearch efficiently stores and indexes all types of data in a manner that enables fast searches. With multiple Elasticsearch nodes in a cluster, stored documents are distributed across the cluster and can be accessed immediately from any node. Elasticsearch balances multi-node clusters to provide scale and high availability. *A network load balancer should not be used with this cluster*, as Morpheus manages and distributes Elasticsearch requests across the cluster.

An Elasticsearch index is a logical grouping of one or more physical shards, where each shard is an index. Each index belongs to one primary shard and one or many replica (redundant copies) shards. Elasticsearch provides redundancy by distributing the documents in an index across multiple shards and distributing those shards across multiple nodes. By default, Morpheus creates 1 index per day for the activity feed, 1 index per day for logs, 1 index per day for monitor check results, and 1 index per day for stats. By default, 1 replica is configured per index.

ℹ When deploying a distributed architecture, Morpheus recommends customers utilize the Elasticsearch installer with the embedded Java option.

Messaging Tier

The Messaging tier is an AMQP-based tier using RabbitMQ for queue services. The RabbitMQ component of Morpheus provides messaging capabilities within the application. RabbitMQ can run as a single instance. For high availability, a cluster of at least 3 nodes is required.

High Availability (HA) for the Messaging Tier

Minimally, a three-node RabbitMQ cluster should be established to accomplish single-site high availability. RabbitMQ clustering and queue mirroring are intended to be used across a LAN – clustering and mirroring across WAN sites is not recommended.

Configuration alignment across RabbitMQ nodes should include compatible versions of RabbitMQ and Erlang, similar installation locations, firewall settings, hosts file, erlang cookie file, and start-on-boot configurations.

Care should be taken to ensure version consistency with the respective Morpheus release. When upgrading the Morpheus Data Appliance, any applicable upgrades to the RabbitMQ must be performed *prior* to performing the Morpheus upgrade. *A network load balancer should not be used with this cluster*. Morpheus manages and distributes RabbitMQ requests across the cluster.

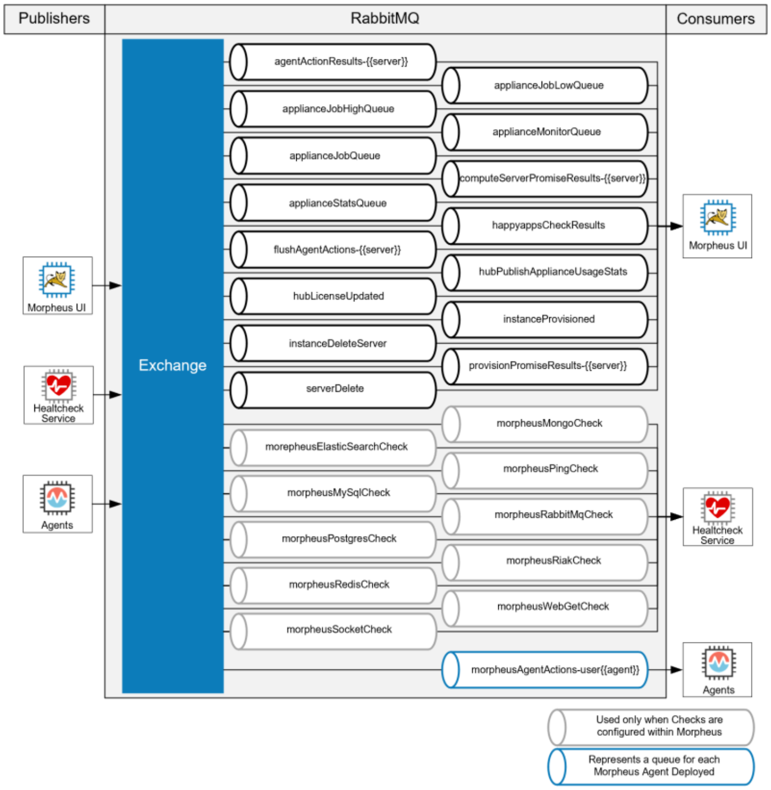
RabbitMQ clusters tolerate the failure or shutdown of individual nodes which can be restarted and rejoin the cluster, as long as the nodes can contact a cluster node within five minutes of boot. Nodes in a RabbitMQ cluster are equal peers. RabbitMQ nodes are identified by unique node names that are appended to hostnames which must be resolvable. RabbitMQ nodes share a secret called *the erlang cookie* in order to communicate. A copy of the erlang cookie is needed at *\home* of root and for each non-privileged user that intends to use CLI tools.

By default, message queues are reachable from all nodes. However, queue mirroring should be enabled to allow queue contents to be replicated across nodes. Each mirrored queue includes a *queue master* - operations are applied to the queue master and then propagated to queue mirrors. Queue mirroring enhances *availability* but not *load distribution*, since each node must perform the same operations. If the queue master fails, the oldest synchronized mirror is promoted to queue master.

ⓘ Morpheus recommends the use of only disk nodes.

Queue mirroring is configured by setting policies. Policies match queues by name using regular expressions. Details about these policies and how to configure them can be found within the [How to set recommended 3-node RabbitMQ policies]() Knowledge Base article. The image below shows the publishers and consumers of Morpheus queues.

Figure 1. Message Queues



Transactional Database Tier

The Transactional Database Tier consists of a MySQL compatible database. The MySQL component of Morpheus provides a logical data store. If redundancy is desired, it is recommended that a *lockable* clustered configuration be used.

High Availability (HA) for the Transactional Database Tier

When high availability is required, Morpheus recommends deploying a Percona XtraDB cluster for the Transactional Database Tier. A Percona XtraDB cluster provides a highly available, scalable, and tightly coupled database cluster. The cluster consists of three nodes, where each node contains the same set of data synchronized across all nodes. These nodes must physically reside close to each other and cannot be geographically diverse. *A network load balancer should not be used with this cluster*. Morpheus manages the load balancing of the Transactional Database Tier and can be configured for either failover or load balancing of the cluster's nodes.

Percona XtraDB Cluster integrates *Percona Server for MySQL* running with the *XtraDB storage engine*, and *Percona XtraBackup with the Galera library* to enable multi-master replication. In multi-master replication, any node can accept write operations; the write will be applied to all nodes in the cluster. This type of replication is called *virtually synchronous replication* because the primary may apply events faster than the secondary nodes. This cluster is capable of scaling read workload as any node can provide read data; however, each node must write all the same data. Percona XtraDB Clusters can tolerate loss of a node without losing data, as long as at least one node remains healthy.

A quorum vote is executed when a cluster node is suspected of being down; the size of the cluster is used to determine the required votes to achieve quorum. Write operations are blocked for a configurable *no-response-timeout* setting, for the duration slightly longer than the configured timeout value when a does not go down gracefully.

To add split-brain protection in cases where the addition of a third node is not possible, an arbitrator node may be deployed. An arbitrator does not store any data or run *mysqld* but is a voting member of the cluster and replicates data.

By default, the Percona XtraDB Cluster uses State Snapshot Transfer (SST) methods of *mysqldump*, *rsync*, or *extrabackup* when all data is copied from one node to another. Using *mysqldump* or *rsync* requires a read-only lock on the database. When only incremental changes are copied from one node to another, the Percona XtraDB Cluster uses Incremental State Transfer (IST). IST is implemented using a caching mechanism on each of the nodes. A *ring-buffer* of last changes is configured and the node is able to transfer this cache. If the number of changes exceeds the ring-buffer the joining node must perform SST.

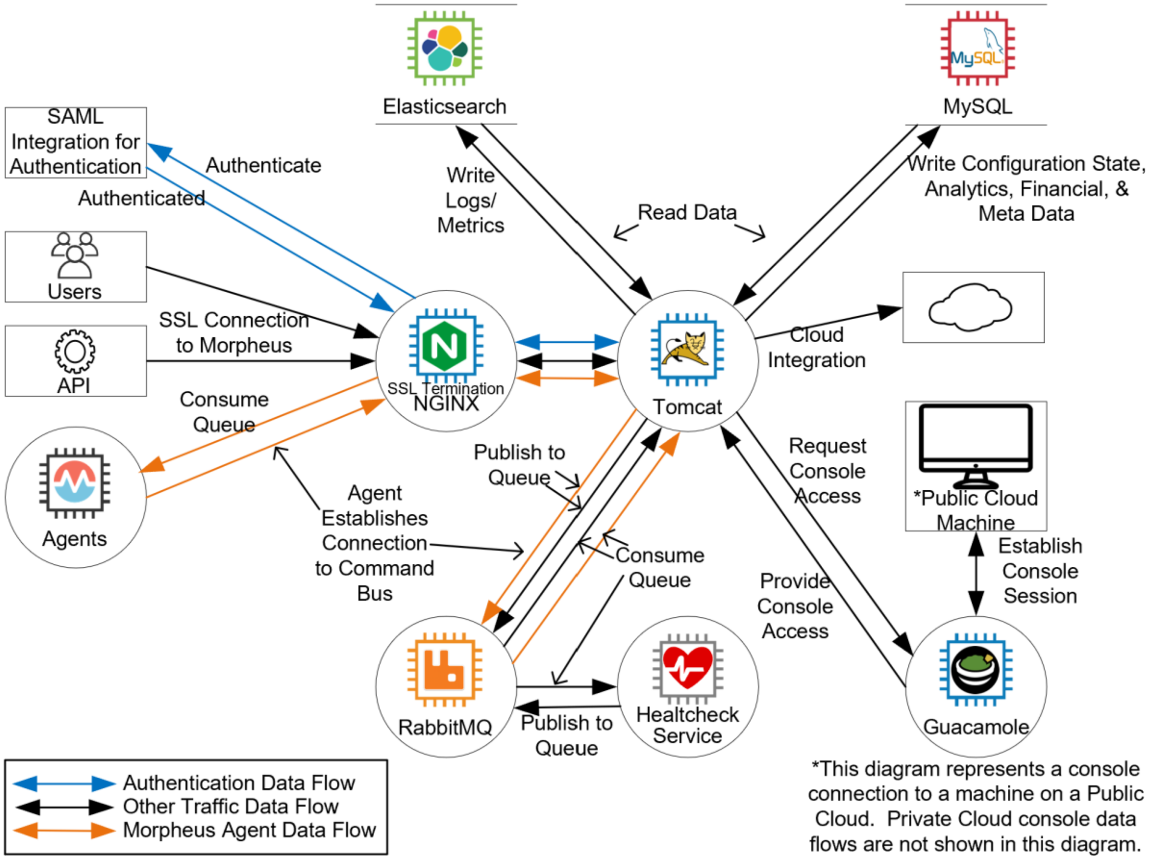
ℹ Avoid creating a cluster of an even number of nodes, as this can lead to a split-brain situation.

ℹ The choice to utilize database software other than that shipped with Morpheus (MySQL) is up to the customer. It is the customer's responsibility to maintain their own database software regardless of the recommendations for clusters contained within this document.

DATA FLOW BETWEEN SERVICE TIERS

The diagram below depicts communication between the difference service tiers of the Morpheus Data Appliance.

Figure 2. Diagrammed Data Flow Between Morpheus Data Appliance



*This diagram represents a console connection to a machine on a Public Cloud. Private Cloud console data flows are not shown in this diagram.

SUPPORTED MORPHEUS DATA APPLIANCE ARCHITECTURES

The architectures described below are supported by Morpheus Data. Deviation from these architectures may result in unsupported configurations or unexpected performance. Morpheus Data professional services should be consulted prior to deviating from any of the below architectures.

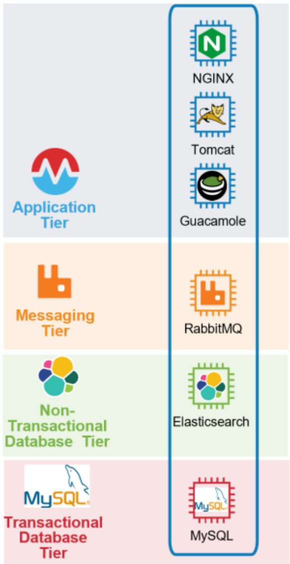
Table 1. Minimum Server Count Per Architecture

Architecture	Combined Tiers	Distributed Tiers
Single Server Architecture	1 Server	
Single Site Redundant Architecture	3 Servers	11 Servers
Multi-Site AWS Multi-AZ Architecture		3 to 6 EC2 instances in addition to Amazon Services
Disaster Recovery Active/Passive Architecture	2 to 22 Servers depending on selected options	

ℹ Refer to *Appendix A of this document* for Morpheus Server resource sizing recommendations.

Single Server (All-in-One) Architecture

Figure 3. Single Server Architecture



Morpheus Single Server (All-in-One) architecture is supported and recommended in all environments where *application-level redundancy* is not required. Many production environments implement *infrastructure-level high availability* to mitigate some of the risks introduced associated running without application-level redundancy.

This architecture has been tested and proven to support up to 5,000 managed workloads with *the Morpheus Data Agent* installed. The maximum achievable number of managed machines may vary based on specific environment configurations and integrated services.

This architecture consists of hosting all tiers of the Morpheus Data Appliance on a single server. This simplifies installation, ongoing maintenance, and troubleshooting. Connectivity between service tiers is not dependent on the underlying network. All service tiers are updated concurrently via the Morpheus installation and update package; however, downtime during upgrades is required. This architecture may be scaled vertically by adding additional CPU and/or RAM, or scaled horizontally by moving to one of the redundant architectures described in the sections below.

Single-Site Redundant Architectures

Single-site architectures can be scaled both vertically and horizontally, allowing customers to achieve high availability and fault tolerance assuming all underlying hardware and dependencies are free of single points of failure. Connectivity between appliance services is dependent on the underlying network, which could potentially cause application instability or performance issues if it is unreliable.

Load balancing and shared storage is required for the Application Tier.

Clusters are required for each of the Non-Transactional Database, Messaging, and Transactional Database tiers. These clusters are sensitive to network interruptions and should be maintained with an odd number of nodes to mitigate split-brain scenarios.

When performing upgrades or maintenance activities, care should be taken to ensure all servers are running the same software version of the Morpheus Appliance as well as the same and compatible versions of the underlying software and operating system patches.

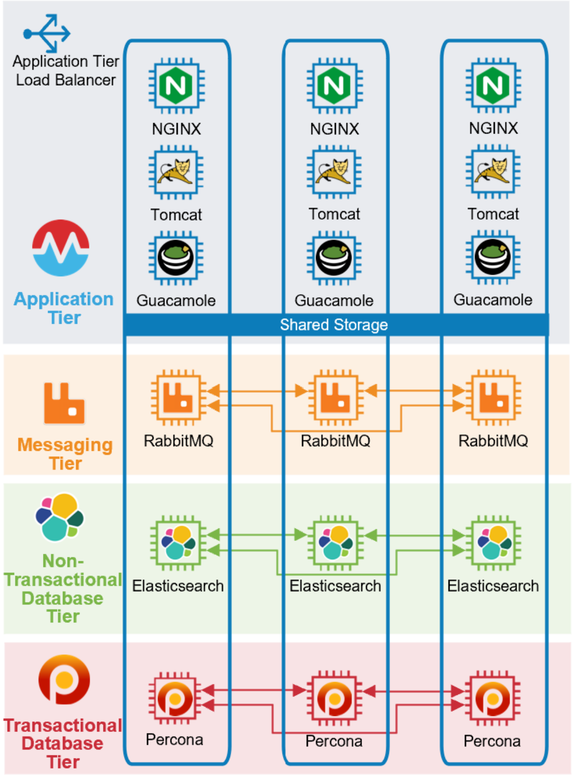
- ℹ These redundant architectures do not require downtime during upgrades; however, the Morpheus installer will only upgrade software components that reside on the same server as the Application Tier. Upgrades of distributed components are not supported by the via the Morpheus installation and update package. Upgrading these components is the responsibility of the customer.

- ℹ If deployed on resource managed infrastructure such as vSphere Distributed Resource Scheduler (DRS), it is recommended that anti-affinity rules be created to ensure redundant servers are not allowed to run on the same host.

Redundant Combined Tiers (3-Node Architecture)

The Morpheus Redundant Combined Tiers architecture (3-Node Architecture) consists of hosting all Morpheus Data Appliance service tiers on each of three servers configured with a network load balancer configured for the Application Tier and a discreet cluster configured for each of the Messaging, Non-transactional Database, and Database tiers.

Figure 4. Redundant Combined Tiers (3-Node Architecture) Conceptual Diagram



Establishing clusters for each of the Messaging, Non-Transactional Database, and Transactional Database tiers across a set of three servers does introduce complexity for ongoing maintenance and troubleshooting.

In this architecture, Morpheus should be configured for failover of the Transactional Database Tier. This enables pointing to a single database server and failing over if it becomes unavailable. Optionally, the Transactional Database Tier may be configured on separate servers.

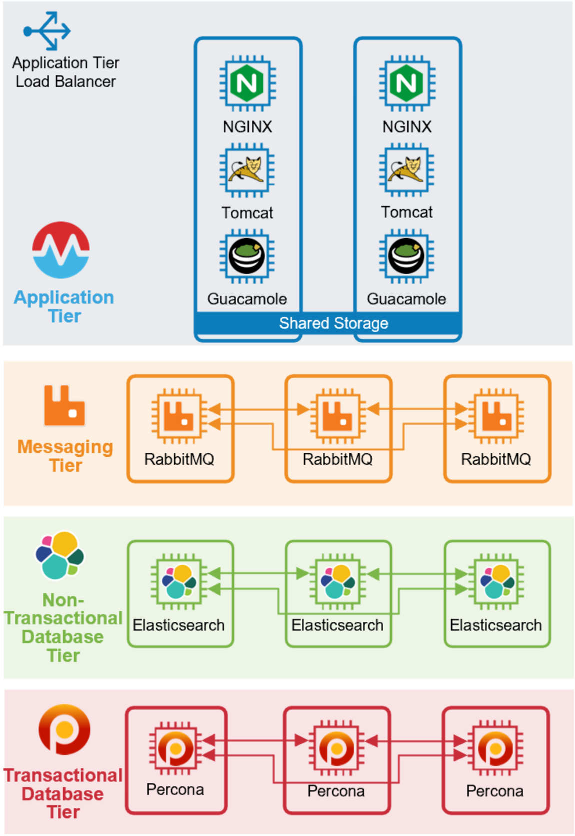
This architecture supports vertical scaling. Scaling horizontally requires moving to a *Distributed Tiers architecture*.

Redundant Distributed Tiers (11-Node+ Architecture)

The Morpheus Redundant Distributed Tiers architecture consists of distributing each appliance service tier onto a different server, where and at least two servers are dedicated to the Application Tier and at least three like-servers are dedicated to each of the Non-Transactional Database Tier, Messaging Tier and Transactional Database Tier clusters.

The distribution of tiers provides the ability to accommodate traditional data center network segmentation. This architecture can be scaled both vertically and horizontally at a single site. When scaling horizontally, an odd number of cluster nodes should be maintained to mitigate split-brain conditions.

Figure 5. Redundant Distributed Tiers (11-Node Architecture) Conceptual Diagram



Morpheus Data Appliance on AWS Single Server (All-in-One) Architecture

Morpheus Data supports the Single Server (All-in-One) architecture in AWS EC2, as long as the operating system of the EC2 instance is supported. See the *Deployment Requirements* section of this document for supported operating systems.

Morpheus Data Appliance on AWS with Redundant Tiers

The Morpheus Data on AWS with Redundant Tiers architecture is supported in all environments where AWS services are available, and is essentially the *Single Site Redundant Distributed Tiers architecture* deployed in a manner that spans multiple AWS Availability Zones (AZs).

- ℹ Morpheus Data recommends deploying to an AWS with at least three AWS AZs, to provide a higher degree of fault tolerance. Morpheus Data will support a deployment to an AWS Region with two AZs (e.g. AWS NorCal), if the architecture includes three instances where at least one of the instances is connected to a discreet AZ.

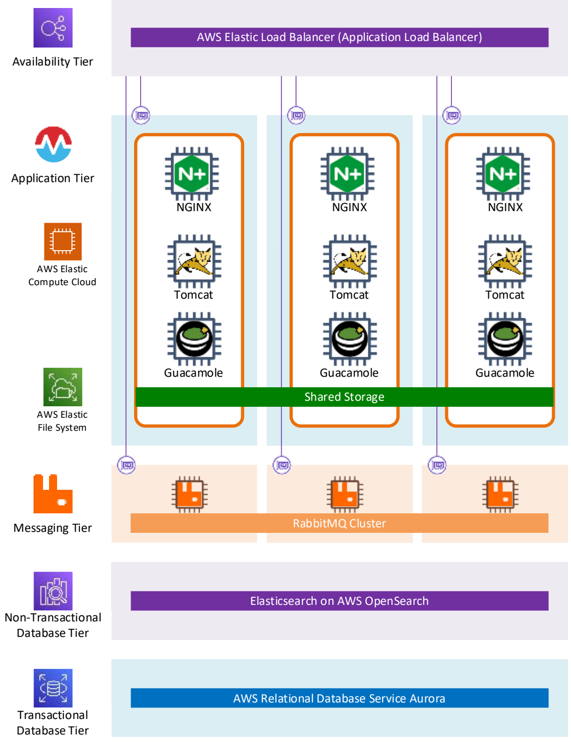
- ⚠ DNS name resolution must be available for all nodes and managed machines.

- ⚠ Spanning AWS Regions is not supported. Spanning AWS VPCs is not supported.

In this architecture, the Amazon Elastic Load Balancer (ELB) service provides high availability for the Application Tier, where the Amazon Elastic Compute Cloud (EC2) service provides cloud VM instances and the AWS Elastic File System (EFS) service provides shared storage.

RabbitMQ, deployed on Amazon EC2 instances, provides AMQP services for the Messaging Tier. Amazon OpenSearch (Elasticsearch) is used for the Non-Transactional Database tier, and Amazon RDS (Aurora) is used for the Transactional Database Tier.

Figure 6. AWS Multi-Availability Zone Conceptual Diagram


Availability Tier
AWS Elastic Load Balancer (Application Load Balancer)
Application Tier
NGINX
NGINX
NGINX
AWS Elastic Compute Cloud
Tomcat
Tomcat
Tomcat
Guacamole
Guacamole
Guacamole
AWS Elastic File System
Shared Storage
Messaging Tier
RabbitMQ Cluster
Non-Transactional Database Tier
Elasticsearch on AWS OpenSearch
Transactional Database Tier
AWS Relational Database Service Aurora

Application Tier

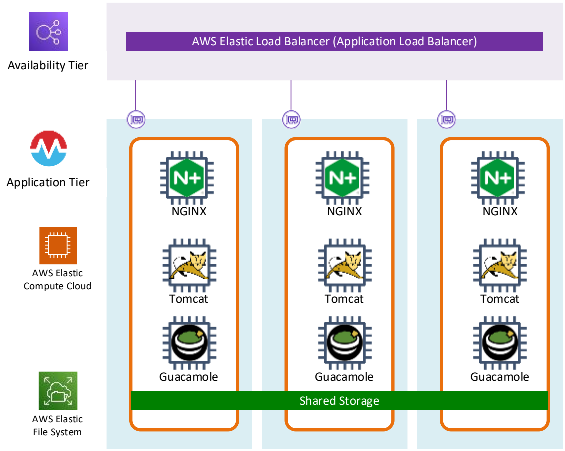
Three EC2 instances should be deployed to support the Application tier, each with an Elastic Network Interface (ENI) attached to a discreet AZ. Morpheus Data will support a deployment to an AWS Region with only two AZs, but recommends a different AZ per ENI.

An AWS ELB Application Load Balancer (ALB) should be configured with an ALB Listener for incoming traffic configured as the Appliance URL of the Morpheus Data Appliance.

- A valid SSL certificate will be required for the ALB Listener.
- The Target Group of the ALB should contain the EC2 instances of the Morpheus Data Appliance nodes.
- The Listener of the ALB should be configured for group-level stickiness on the Target Group.
- The Target Group should also be configured for Stickiness with load balancer generated cookie. The Morpheus Data Appliance does not provide application-level cookies for sticky sessions.
- The Network Mapping of the ALB should be configured as the subnet to which each ENI of each Application tier server is attached.
- The load balancing algorithm of the ALB should be configured *Least Outstanding Requests* with session persistence enabled.

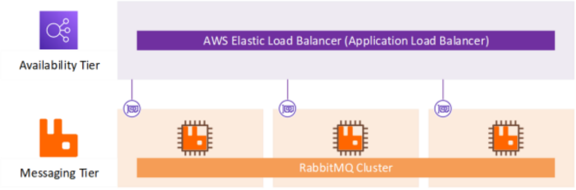
Amazon Elastic File System (EFS) shared storage should be configured to support each Application tier server. Objects stored in EFS will include white label PNG files, uploaded virtual images, Ansible playbooks, Terraform plans, and Morpheus Data Appliance backup files.

Name resolution, NTP, and authentication solution connectivity should be available for all systems from all AWS AZs.

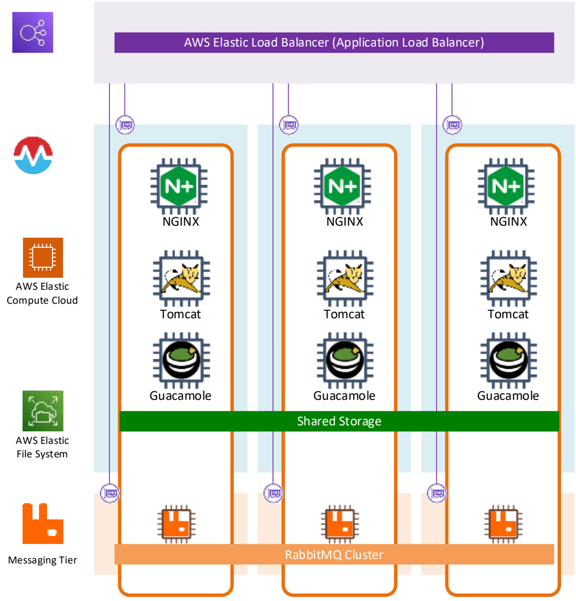


Messaging Tier

A three-node cluster of RabbitMQ instances should be deployed in AWS EC2, with each instance's ENI attached to a discreet AZ. The three RabbitMQ node ENIs should be attached to each of the same AZs as the Application tier server ENIs.



Optionally, the Application tier and the Messaging tier can be combined on the same EC2 instances.



Non-Transactional Database Tier

A new domain in AWS OpenSearch should be created to support for the Non-Transactional Database tier. The domain should be of type Elasticsearch and span the same Availability Zones as the Application tier and the Messaging tier. Enabling domain auto-tune is recommended but not required.

Morpheus recommends enabling node-to-node encryption in the OpenSearch domain. Enabling node-to-node encryption in the OpenSearch domain enables TLS 1.2 encryption for Elasticsearch communications within the AWS Virtual Private Cloud (VPC).

⚠ Node-to-node encryption must be configured at the time of domain creation, it is not possible to configure node-to-node encryption on existing domains. Node-to-node encryption cannot be disabled after it is configured.

ℹ It is the customer's responsibility to ensure version compatibility is maintained between Morpheus and Amazon ES.

Transactional Database Tier

A new Amazon Aurora database instance should be configured, in the Amazon Relational Database (RDS) service, to support the Transactional Database tier.

The new Amazon Aurora database instance must be configured to use the MySQL-compatible engine for MySQL version 5.7 or later.

⚠ At time of publication of this document, the Morpheus Data Appliance requires MySQL version 5.7 or later in the Transactional Database tier. Amazon Aurora version 1.x uses MySQL version 5.6, which is not compatible with the Morpheus Data Platform. Amazon Aurora version 2.04.0 or later should be used.

Morpheus Data does not provide any recommendation on credential management for the Amazon Aurora database instance. Credential management for the Amazon Aurora database instance is at the discretion of the customer.

The Amazon Aurora instance should include at least 4 vCPUs and 16 GiB of RAM.

Morpheus Data recommends configuring an Amazon Aurora Replica.

EC2 connections should be created in RDS for each EC2 instance in the Application Tier of the Morpheus Data appliance. AWS Security Groups applied to the EC2 instances must allow communication from the EC2 instance to the Amazon Aurora database instance.

The Amazon Aurora instance should use an RDS Subnet Group which includes all AZs and subnets to which the Application Tier instance’s ENIs are attached.

The Amazon Aurora instance must be created in the same VPC as the Morpheus Data Appliance EC2 instances.

Morpheus Data does not provide any recommendation on configuring an RDS Proxy. RDS Proxy configuration, if any, for the Amazon Aurora database instance is at the discretion of the customer.

Morpheus Data does not provide any recommendation on maintenance window configuration for the Amazon Aurora database instance. Maintenance window configuration for the Amazon Aurora database instance is at the discretion of the customer.

Morpheus Data recommends disabling Backtrack for Amazon Aurora in all Morpheus deployments. Asynchronous database rollbacks in the Transactional Database tier are likely to cause incongruence between the Morpheus Data Appliance and any managed cloud environments.

Morpheus Data recommends disabling auto-minor-version-upgrade on the Amazon Aurora instance. See the online Morpheus documentation at https://docs.MorpheusData.com, or check with your Morpheus Data support and engineering teams, for the latest version compatibility information.

ℹ It is the customer's responsibility to ensure version compatibility is maintained between the Morpheus Data Appliance and Amazon Aurora.

Morpheus Data highly recommends enabling Deletion Protection on the Amazon Aurora database instance.

DISASTER RECOVERY

There are several options available regarding disaster recovery and Morpheus. Customers may choose to simply restore a database backup to a warm standby architecture at a secondary site or they may chose more complex options based on their RPO, RTO, and that of the underlying technologies and dependent services. The architecture below shows several options for customers interested in disaster recovery planning.

Multi-Site Active/Passive Architecture

Morpheus Multi-Site Active/Passive Architecture is supported in all environments where a secondary failover site is required. This architecture provides the capability to failover and is intended only for disaster recovery use cases. With this architecture, users must never point to both locations at the same time. Failover and failback are not automated, given the disaster recovery use case it is expected that a specific decision must be made to execute a manual failover, and configuring failback would be a manual process as well that would occur only after the threat has been neutralized.

Note: Automated failover and failback is not a capability currently on the Morpheus roadmap.

This architecture allows customers to achieve only failover capability of Morpheus, access to underlying technologies and dependent services is required for Morpheus to function properly. This architecture requires both cross-region shared storage and cross-region load balancing, as well as, access to underlying and dependent data center services. With this architecture consistently stable WAN connectivity is important to enable a viable failover solution. When performing upgrades and/or maintenance care should be taken to ensure all servers in all regions are running the same version of Morpheus as well as the same version of component software and operating system patches.

Option: Customers may choose different or similar single site architectures for both the primary and failover sites. They may even choose to accept a reduction in availability and capability at the failover site as compared to those offered at the primary site. Operational and availability requirements during a disaster event should be fully understood when choosing the architecture and defining the capabilities of the failover site. In any case, care should be taken to ensure the failover site maintains enough resources for the replicated Morpheus data.

Note: Follow all recommendations in the selected single site architecture sections above in addition to the recommendations within this section.

Application Tier

Morpheus recommends two web servers deployed at each location minimally in order to establish high availability for this tier at each location. Cross-region load balancers and cross-region shared storage must be properly configured to ensure each region's Application Tier operates as expected. Additionally, name resolution, NTP, authentication solution, and integration connectivity should be available for all systems across regions.

LOAD BALANCER

A cross-region load balancer should be configured for active/passive failover between sites. The load balancer should direct users to a single site at a time. During a failover event, the load balancer should redirect users to the failover site after replication from the primary site has been halted, never allowing customers to access both sites at the same time.

Non-Transactional Database Tier

Elasticsearch Cross-Cluster Replication (CCR) available in version 6.7 or higher enables pull-based (driven by the follower) replication of indices from one Elasticsearch cluster to another. Morpheus supports Elasticsearch 7.x in Morpheus 4.2 or later releases. This technology replicates data between two distinct clusters, in order to provide high availability at both locations follow the recommendations in the Redundant Architectures section to establish a cluster at each location prior to configuring CCR. Cross-cluster replication works by replaying the history of individual write operations that were performed on the shards of the leader index. Soft deletes occur whenever an existing document is deleted or updated, by retaining these soft deletes on leader shards they are made available for replay. CCR is active-passive since the leader index can be written directly and the follower index cannot. The leader is capable of accepting index writes and the followers have read-only copies of the index. When a leader index is not available, another index must be explicitly chosen for writes by the cluster administrator.

Note: CCR requires the purchase of an Elasticsearch Platinum Level subscription.

Option: Customers may choose not to replicate Elasticsearch data to the failover site. This will result in the loss of historical metrics, stats, and logs collected from machines (logs would still exist on the individual machines). Cost information would not be lost as it is stored in the SQL database.

Messaging Tier

Of the two replication plugins available the Federation plugin is the only one supported by Morpheus. Do not use the Shovel plugin with Morpheus.

The Federation plugin allows an exchange or queue in one cluster to receive messages published to an exchange or queue of another. Exchange federation links will start on any node in the downstream cluster and will failover to other nodes in case of an outage. The Federation plugin is designed to tolerate intermittent communication and aims to provide opinionated distribution of exchanges and queues using the erlang client. The Federation plugin communicates via the Erlang AMQP client and is included with the RabbitMQ distribution. Because this technology connects two distinct clusters, in order to provide high availability at both locations follow the recommendations in the Redundant Architectures section to establish a cluster at each location prior to configuring the Federation plugin.

Option: Customers may choose not to replicate RabbitMQ data to the failover site. This will result in the loss of any queued actions. The effects of this would be minimal and would typically go unnoticed in most environments.

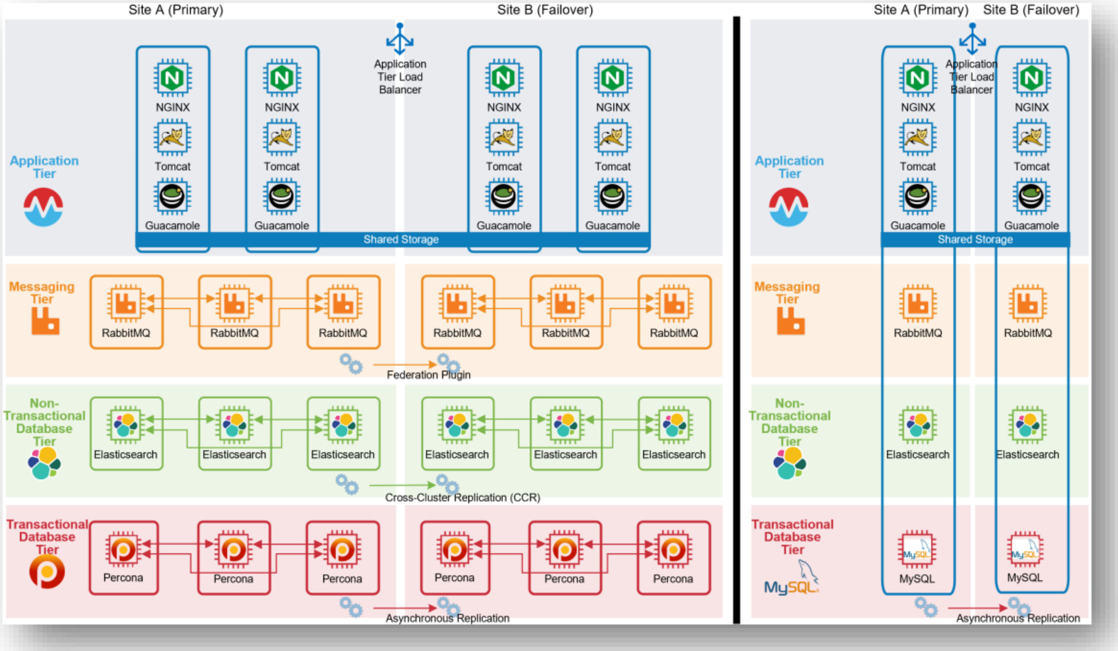
Transactional Database Tier

Morpheus recommends using standard MySQL asynchronous replication to span geographic locations establishing loosely coupled database clusters. Within the Transactional Database Tier section of this document, a tightly coupled database cluster utilizing the Galera library to configure multi-master replication is described for single-site high availability. Tightly coupled database clusters and/or clusters using multi-master replication are not recommended to span geographic locations. Given that the failover site is geographically separated by a presumed significant distance with WAN connectivity the only viable solution is to use asynchronous replication. Asynchronous replication allows for independence in processing and applying each transaction.

Because this technology connects two distinct clusters, in order to provide high availability at each location, follow the recommendations in the Redundant Architectures section to establish tightly coupled database clusters at each location prior to configuring asynchronous replication. It is possible to shift from one node to another inside either of the two clusters, while the asynchronous data continues passing from one source to the other. If using Percona ExtraDB, use the Replication Manager to ease configuration and management.

The image below shows a couple of sample conceptual diagrams. Depending on the options single site architectures chosen for each location the diagram may look quite different.

Figure 7. Disaster Recovery Conceptual Diagram SAMPLES



APPENDIX A: MORPHEUS SERVER SIZING

Table 2. Morpheus Server Sizing for a Single-Node All-in-One Appliance

Combined Tiers	
Memory	16 GB
CPU	4 Core 1.4 GHz (or better) CPU
Non-Redundant Architectures Storage	Minimum 200 GB (Local Storage) - Morpheus binaries, virtual images, backups, logs, stats, user uploaded, and user imported data require adequate space on the Morpheus Server. Recommended Swap Disk - .5 times RAM if 16 GB or 1 times RAM if 8 GB.
Redundant Architectures Storage	Minimum 50 GB (Local Storage) - Morpheus binaries Recommended Swap Disk - .5 times RAM if 16 GB or 1 times RAM if 8 GB. Minimum 200 GB (Shared Storage) - Virtual images, backups, logs, stats, user uploaded, and user imported data require adequate space on the shared file system.

Table 3. Morpheus Server Sizing for Distributed Architecture Nodes

Distributed Tiers	
Application Tier	
Memory	16 GB RAM
CPU	2 Core 1.4 GHz (or better) CPU
Non-Redundant Architectures Storage	Minimum 200 GB (Local Storage) - Morpheus binaries, virtual images, backups, logs, stats, user uploaded, and user imported data require adequate space on the Morpheus Server. Recommended Swap Disk of 0.5 times RAM
Redundant Architectures Storage	Minimum 50 GB (Local Storage) - Morpheus binaries Recommended Swap Disk of 0.5 times RAM Minimum 200 GB (Shared Storage) - Virtual images, backups, user uploaded, and user imported data require adequate space on the shared file system.
Non-Transactional Database Tier	
Memory	Recommended 8 GB RAM
CPU	2 Core 1.4 GHz (or Better) CPU
Storage	Recommended 150 GB (Local Storage) - Elasticsearch binaries for Elasticsearch servers. Recommended Swap Disk - 1 times RAM

Messaging Tier	
Memory	Recommended 8 GB RAM
CPU	2 Core 1.4 GHz (or Better) CPU
Storage	Recommended 150 GB (Local Storage) - RabbitMQ and Erlang binaries Recommended Swap Disk - 1 times RAM
Transactional Database Tier	
Memory	Recommended 8 GB RAM
CPU	2 Core 1.4 GHz (or Better) CPU
Storage	Recommended 200 GB - Morpheus database, Percona binaries Recommended Swap Disk - 1 times RAM

ⓘ The swap size recommendation is .5 times the amount of RAM. Customers may choose to follow their standard enterprise swap sizing best practices.

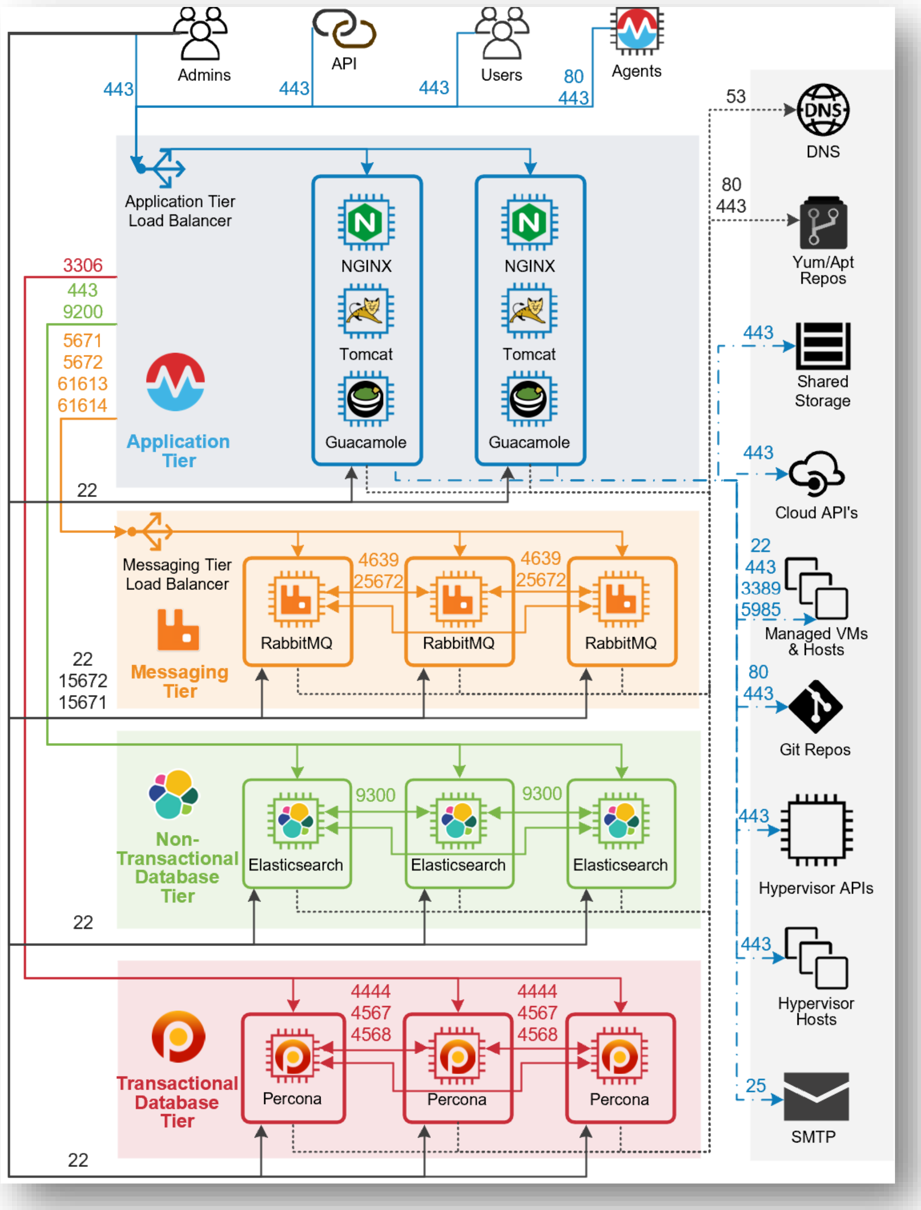
APPENDIX B: PORTS AND PROTOCOLS

Table 4. Ports and Protocols

Source	Destination	Port	Protocol	Description
User	Application Tier	443	TCP	User Access
Morpheus Servers	DNS Servers	53	TCP	Domain Name Resolution
Morpheus Servers	Time Source	123	TCP	Time Resolution
Morpheus Servers	Web or Offline Installer	80, 443	TCP	Download repos and Morpheus packages (yum/apt repos)
Managed Machine	Application Tier	443	TCP	Morpheus Agent Communications
Managed Machine	Application Tier	80, 443	TCP	Agent Installation. (Requires port 80 only for Ubuntu 14.04)
Managed Machine	Application Tier	N/A	N/A	Agent Installation Clout-init (Linux)
Managed Machine	Application Tier	N/A	N/A	Agent Installation Cloudbase-init (Windows)
Managed Machine	Application Tier	N/A	N/A	Agent Installation VMtools
Managed Machine	Application Tier	N/A	N/A	Static IP Assignment & IP Pools (Cloud-init or VMware Tools)
Managed Machine	Docker Image Repo	443	TCP	Applicable if using docker
Managed Machine	Application Tier	69	TCP/UDP	PXE Boot (Forwarded to internal PXE port 6969)
Application Tier	Managed Machine	5985	TCP	Agent Installation WinRM (Windows)
Application Tier	Managed Machine	22	TCP	Agent Installation SSH (Linux)
Morpheus Application Tier	Managed Machine	22, 3389, 443	TCP	Remote Console (SSH, RDP, Hypervisor Console
Application Tier	AWS S3	443	TCP	Morpheus Catalog Image Download
Application Tier	Hypervisor	443	TCP	Hypervisor hostname resolvable by Morpheus Application Tier
Application Tier	Non-Transactional Database Tier	443	TCP	Applicable if using Amazon Elasticsearch Service
Application Tier	Docker CE Repo	443	TCP	Applicable only when integrated with Docker
Application Tier	Rubygems	443	TCP	
Application Tier	Morpheus Hub	443	TCP	(Optional) Telemetry data (Disabled only via license feature)

Application Tier	Mail Server	25 or 465	SMTP	Send email from Morpheus
Application Tier	Messaging Tier	5672	TCP	AMQP non-TLS connections
Application Tier	Messaging Tier	5671	TCP	AMQPS TLS enabled connections
Application Tier	Messaging Tier	61613	TCP	STOMP Plugin connections (Required only for Morpheus versions 4.2.1 or prior)
Application Tier	Messaging Tier	61614	TCP	STOMP Plugin TLS enabled connections (Required only for Morpheus versions 4.2.1 or prior)
Messaging Tier	Messaging Tier	25672	TCP	Inter-node and CLI tool communication
Administrator Web Browser	RabbitMQ Server Management	15672	TCP	Management plugin
Administrator Web Browser	RabbitMQ Server Management	15671	TCP	Management plugin SSL
Messaging Tier Cluster Node	Messaging Tier Cluster Node	4369	TCP	erlang (epmd) peer discovery service used by RabbitMQ nodes and CLI tools
Application Tier	Non-Transactional Database Tier	9200	TCP	Elasticsearch requests (Used in all cases except when utilizing AWS ES service)
Non-Transactional Database Tier	Non-Transactional Database Tier	9300	TCP	Elasticsearch Cluster
Transactional Database Tier	Transactional Database Tier	4567	TCP/UDP	Write-set replication traffic (over TCP) and multicast replication (over TCP and UDP).
Transactional Database Tier	Transactional Database Tier	4568	TCP	Incremental State Transfer (IST)
Application Tier	Transactional Database Tier	3306	TCP	MySQL client connections
Backup Solution	Transactional Database Tier	4444	TCP	State Snapshot Transfer (SST)
Application Tier	Integrated Technology	Varies	TCP	Integrations (Uses the port of the 3rd party systems API)

Figure 8. Port Diagram


Admins
API
Users
Agents
443
443
443
80
443
53
DNS
80
443
Yum/Apt Repos
Application Tier Load Balancer
NGINX
Tomcat
Guacamole
NGINX
Tomcat
Guacamole
Application Tier
3306
443
9200
5671
5672
61613
61614
22
443
Shared Storage
443
Cloud API's
Messaging Tier Load Balancer
4639
25672
4639
25672
RabbitMQ
RabbitMQ
RabbitMQ
Messaging Tier
22
15672
15671
22
443
3389
5985
Managed VMs & Hosts
80
443
Git Repos
9300
9300
Elasticsearch
Elasticsearch
Elasticsearch
Non-Transactional Database Tier
22
443
Hypervisor APIs
443
Hypervisor Hosts
4444
4567
4568
4444
4567
4568
Percona
Percona
Percona
Transactional Database Tier
22
25
SMTP