

Events Service Deployment

On this page:

- [About the Events Service](#)
- [Deployment Topology Overview](#)
- [Single Node Deployment](#)
- [Multi-Node Cluster](#)
- [Supported Deployments](#)
- [Shared Events Service](#)
- [Default Ports](#)

Related pages:

- [Events Service Requirements](#)
- [Install the Events Service on Linux](#)
- [Install the Events Service on Windows](#)

The topic describes the AppDynamics Events Service, the on-premises data storage facility for unstructured data generated by Application Analytics, Database Visibility, and End User Monitoring deployments.

About the Events Service

The MySQL database embedded in the Controller stores application metric and configuration data generated by the Controller. While an embedded database is sufficient for storing this type of data, the high-volume, performance-intensive nature of analytics data requires dedicated, horizontally scalable storage. In an AppDynamics deployment, this role is served by the AppDynamics Events Service.

If you are installing the server components for End User Monitoring, Application Analytics, or Database Visibility, you need to use a scaled-out on-premises Events Service. Scaled-out means that the service is not installed on the platformadmin host. This type of Events Service can be deployed as a single node or a cluster of three or more nodes based on your needs. Additionally, you can add nodes to a scaled-out Events Service after you install it. It is not recommended that you add the Controller host as part of the cluster. For information, see [Administer the Events Service](#).

For data redundancy and storage scalability, however, or if you are using End User Monitoring or Application Analytics in an on-premises Controller deployment, you need to deploy a dedicated Events Service installation.

Multiple AppDynamics components that depend on the Events Service should use the same Events Service instance or cluster.

Deployment Topology Overview

You can deploy the Events Service to a single node or to a cluster of three or more nodes. Clusters are horizontally scalable, so nodes can be added as your data storage requirements grow. Clusters also provide data replication and redundancy, helping to ensure data integrity in the event of a node failure.

The Controller includes an embedded Events Service instance used by the Database Visibility product by default. However, the embedded Events Service is not meant to be used with production Application Analytics or EUM installations, since it runs on the same machine as the Controller and does not offer data replication or scalability. It may be used for small-scale environments, especially those intended for demonstration or learning purposes. Note however that it's not possible to migrate data from the embedded Events Service to a external Events Service instance if upgrading later.

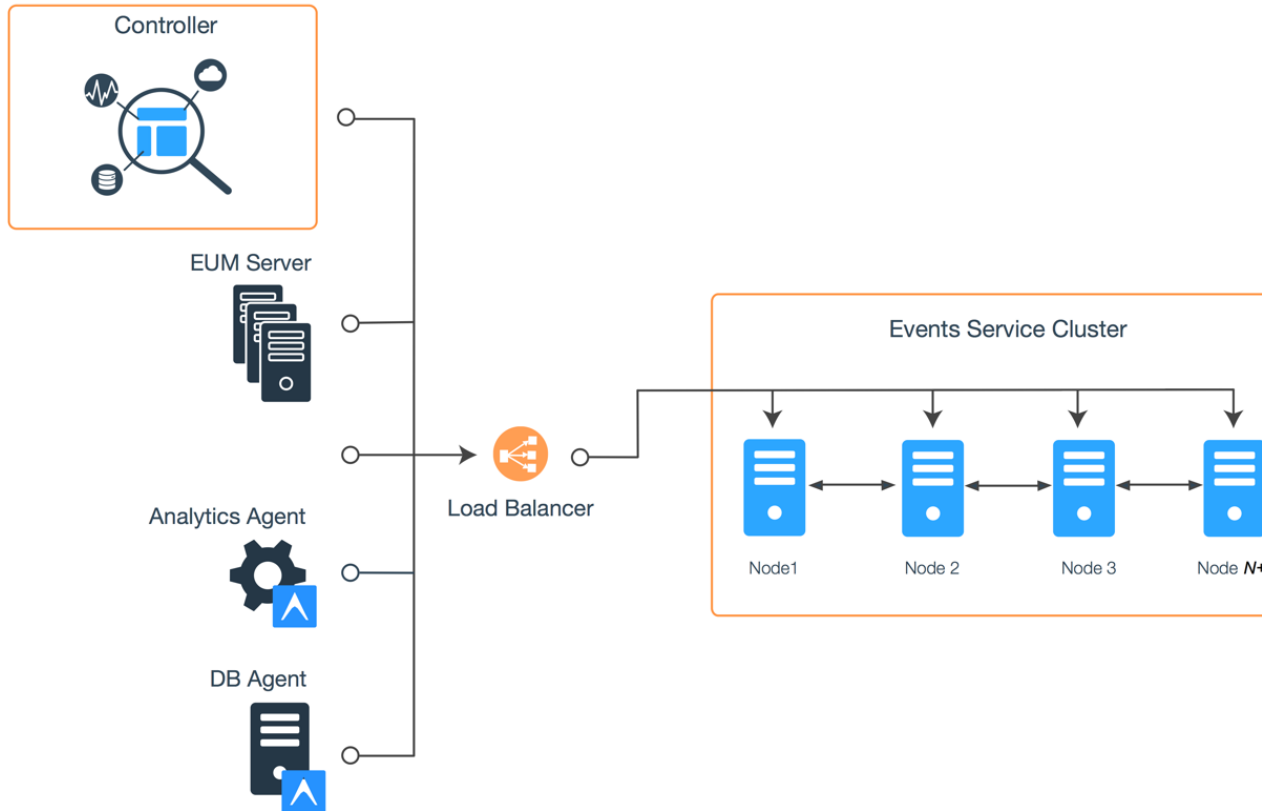
Your Events Service can be deployed to support multiple Controllers, thereby becoming a shared Events Service.

Single Node Deployment

In a single node Events Service deployment, the Events Service runs on a dedicated machine. The Controller and other Events Service clients can connect directly to the Events Service node or through a load balancer. Deploying a single node Events Service behind a load balancer allows you to grow the deployment to a multi-node cluster easily, without having to modify the clients.

Multi-Node Cluster

A multi-node cluster is made up of three or more nodes. With a cluster, the Controller and other Events Service clients, the EUM Server and Analytics Agent connect to the Events Service through a load balancer, which distributes load to the Events Service cluster members.



The nodes in a cluster swap a large amount of data. For this reason, when deploying a cluster, make sure to install all cluster nodes within the same local network, ideally, attached to the same network switch.

Supported Deployments

Use the following table to determine which deployment types and environments are supported for different versions of the Events Service.

Deployment Types	Development Environment	Production Environment
Controller Embedded Events Service (version 4.3.x)	Yes	No. Only shipped through the Controller installer.
Controller Embedded Events Service (version 4.4.x)		
Controller Embedded Events Service (version 4.5.x)		
Single Node Events Service (Dedicated Host) (version 4.3.x)		Yes
Single Node Events Service (Dedicated Host) (version 4.4.x)		
Single Node Events Service (Dedicated Host) (version 4.5.x)		
Multi-Node Clustered Events Service (3, 5, 7+ nodes) (version 4.3.x)		

Multi-Node Clustered Events Service (3, 5, 7+ nodes) (version 4.4.x)	
--	--

Multi-Node Clustered Events Service (3, 5, 7+ nodes) (version 4.5.x)	
--	--

Shared Events Service

You can connect multiple Controllers to a single on-premises Events Service deployment using the same procedure that is required to connect a single Controller. You would need to configure the Controller URLs for the Events Service to point to the shared Events Service, and make sure the Controller keys are correct. The Controllers can then handle syncing to the shared Events Service. There are no additional requirements.

When compared to a multiple Events Service cluster configuration, a shared Events Service configuration requires less maintenance and lowers cost. Since the Events Service is horizontally scalable, having a single large instance provides the same functionality as multiple instances.

There is no limit to the number of Controllers that can share an Events Service. However, it is recommended that you use separate Events Services for dev and prod. Work with your AppDynamics account representative if you plan to expand your shared Events Service cluster.

Default Ports

The default ports used by the Events Service are:

- Events Service API Store Port: 9080
- Events Service API Store Admin Port: 9081

The Events Service cluster members use additional ports for internal communication among the cluster members. All the ports used within the cluster are listed in the Events Service configuration file, `conf/events-service-api-store.properties`.