Deploy AppDynamics in the Enterprise

Planning the Deployment

Deploying AppDynamics to its live operating environment introduces requirements and considerations beyond those applicable to an evaluation installation.

Security, availability, scalability, and performance all play an important role in production deployment planning. The system resources of the machine that hosts the Controller in a live environment must be able to support the expected workload.

For information on installing the Controller and App Agents, see Install and Upgrade AppDynamics.

This topic covers considerations applicable to deploying AppDynamics to its live environment.

Deployment Tasks

A typical deployment of AppDynamics involves these tasks and considerations:

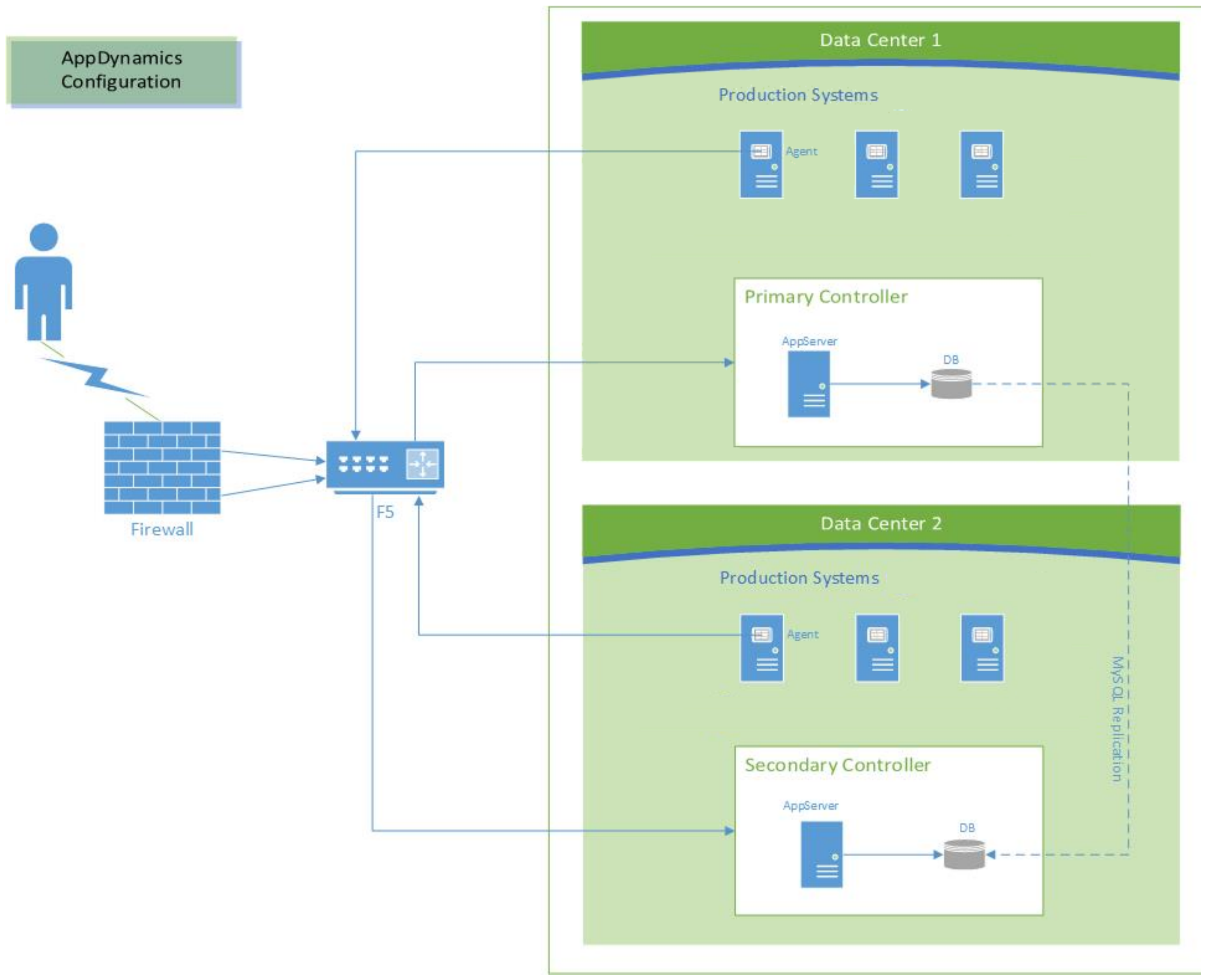
- Ensure that target systems meet the Controller System Requirements for the Controller's expected workload.
- For high volume environments, tune the OS and Controller settings for high workload. Also be sure to verify and if necessary set the file descriptor limits as advised in Configure File Descriptor Limits on Linux.
- Implement Controller High Availability to ensure service continuity in the event of a failure of the Controller server.
- Configure the network environment. If deploying the Controller with a reverse proxies, configure passthrough of Controller traffic. Also note other Network Requirements for the deployment environment.
- Implement security requirements for your environment. If clients will connect to the Controller by HTTPS, install your custom SSL server certificate on the Controller.

- Generate a password management strategy for the built-in system accounts in the Controller and platform.
- Make sure the mail server is properly configured for the Controller in the target environment and define your alerting strategy.
- Devise your backup strategy. A typical backup strategy consists of frequent partial backups with intermittent full backups.
- Plan your configuration maintenance and enhancement strategy. Changes to the configuration should be staged in a non-critical environment, and rolled into the live environment only after thorough testing. The AppDynamics UI offers the ability to export and import configuration settings from various contexts, including from the transaction detection settings, health rules, or the full application.
- Deploying App Agents is likely to be an ongoing task, especially in dynamic environments where monitored systems are regularly taken down and new ones brought up. There are two basic strategies for deploying large numbers of App Agents across a managed environment:
    1. Deploy the agents independently of the application inside the application server. This method ensures that re-deployments of the application do not overwrite the agent deployment.
    2. Integrate deployment of AppDynamics agents into the deployment of applications. This more sophisticated approach requires modifying the existing application deployment automation scripts.

  For details, see:
    - Multi-Agent Deployment for Java
    - Unattended Installation for .NET

Sample Deployment Topology

The following figure shows a sample deployment environment for AppDynamics. This sample illustrates some of the considerations highlighted in Deployment Tasks.



In this example, a primary and a secondary Controller are deployed to separate data centers. Notice that the primary and secondary Controllers maintain data consistency through MySQL replication. An F5 load balancer proxies traffic for the Controllers, including traffic from the App Agents that are deployed to the monitored applications. Alternatively, an Apache HTTP server or an Nginx server can serve as a reverse proxy for the Controller.

Network Requirements

Deploying the Controller often calls for configuration changes to existing network components, such as to firewalls or load balancers in the network. If the Controller will reside behind a load balancer or reverse proxy, you need to set up traffic forwarding for the Controller. You may also need to open ports used by AppDynamics on firewalls or any other device through which traffic must traverse.

The following are general considerations for the environment in which you deploy AppDynamics. See Installation Cheatsheet for Small Environments for other network configuration requirements. For more about configuring a reverse proxy for the Controller, see Deploy with a Reverse Proxy.

Correlation HTTP Header

AppDynamics adds a custom header to traffic in the monitored environment named singularityheader. This header enables AppDynamics to correlate traffic across tiers. It's important to ensure that any load balancer, proxy or firewall in the network between monitored tiers or between the tiers and the Controller preserves the header added by AppDynamics.

Clock Management

To ensure consistent event time reporting across the AppDynamics deployment, the App Agents attempts to synchronize its time with the Controller time.

It does this by retrieving the time from the Controller every five minutes. It then compares the Controller's time to its own local machine's clock time. If the times are different, whether ahead or behind, it applies a time skew based on the difference to the timestamps for the metrics it reports to the Controller.

If, despite the agent's attempt to report metrics based on the Controller time, the Controller receives metrics that are time-stamped ahead of its own time, the Controller rejects the metrics. To avoid this possibility, AppDynamics recommends maintaining clock-time consistency throughout your monitored environment.

Deployment Considerations

The following topics provide much more information about securing your AppDynamics deployment: