



In this scenario, all AppDynamics services run as SaaS and agents are configured to talk to the public SaaS endpoints. By default, we resolve end-user locations using public geographic databases. Optionally, we can use a custom Geo-Server to map internal IP addresses to geographic locations.

Although not strictly required, we recommend using a reverse proxy such as NGINX or Apache for all server-side components.

NOTE: Components must be licensed separately.

Connections

<p>A Traffic: APM/Database Metrics Protocol: HTTP(S) Default Ports: 443 Public Endpoint: <customer>.saas.appdynamics.com</p>	<p>C Traffic: EUM Beacon Data Protocol: HTTP(S) Default Ports: 80/443 Public Endpoint: col.eum-appdynamics.com</p>
<p>B Traffic: Log/Transaction Analytics Event Data Protocol: HTTPS Default Ports: 443 Public Endpoint: analytics.api.appdynamics.com</p>	<p>D Traffic: EUM Geo Resolution Mapping Data Protocol: HTTP(S) Default Ports: 80/443</p>

Key

- 1 GUI users via web browser
 - 2 Applications using the AppDynamics REST APIs
 - 3 Database Agent
 - 4 APM Agents (Java, .NET, Node.js, PHP, Python, Serverless APM for AWS Lambda, Apache Web Server, C/C++ SDK, Go Language, IBM Integration Bus) with Network Visibility
 - 5 Machine Agents (with Analytics Agent, Server Visibility, and/or AppDynamics Extensions)
 - 6 Browser and Mobile RUM Agents for web/mobile/and IoT applications
- SaaS Hosted**
- 7 EUM Synthetic Monitoring Service
 - 8 Events Service
 - 9 EUM Real User Monitoring Service
 - 10 Controller Service
- On-Premises Optional**
- 11 Custom EUM Geo Server (On-Premises, optional)

Data Stores

- APM Configuration and Metrics:
Controller Database (MySQL) + Data Aggregation Service (HBase) [7]
- EUM Configuration and Metrics:
Controller Database (MySQL) + Data Aggregation Service (HBase) [7]
- EUM Events: Events Service: SaaS [8]
- Transaction/Log Analytics: SaaS [8]
- EUM Geo Resolution Data: GeoServer: Optional On-Premises [11]