



Arquitectura AppDynamics e instalación de  
agentes.

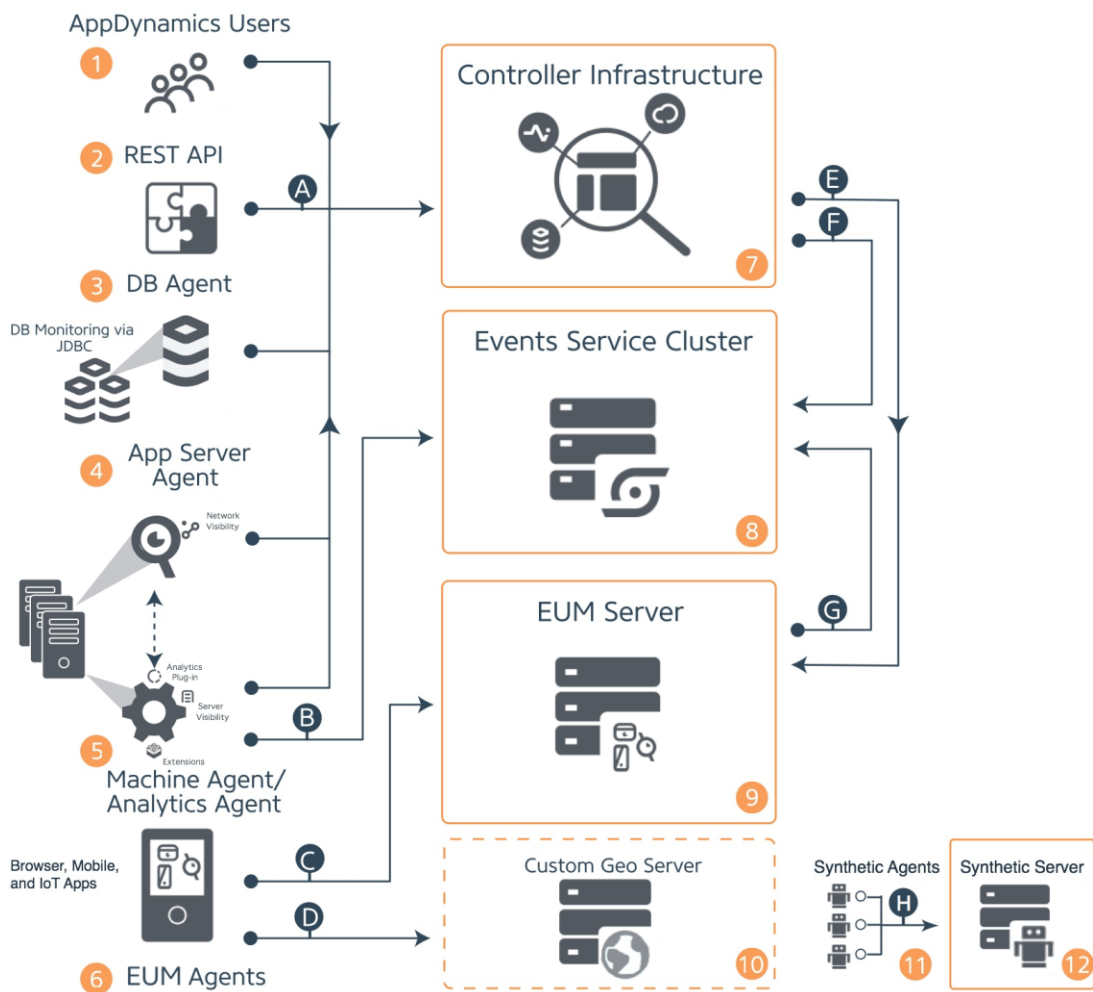
## Contenido

1	Componentes de AppDynamics .....	3
1.1	Controller Infrastructure	4
1.2	Event Server	4
1.3	App Agent Server	4
1.4	Machine Agent	4
1.5	DB Agent	5
2	Enterprise Console .....	6
3	Machine Agent como servicio.....	7
3.1	Descarga del agente	7
3.2	Configuración del archivo Controller-info.xml	9
4	Agentes en Solaris para WebMethods .....	12
5	Alertas configuradas.....	15
6	Referencias .....	17

## 1 Componentes de AppDynamics

Una instalación de AppDynamics Platform local consta básicamente de varios componentes instalados y configurados por separado. Estos incluyen el controlador, la base de datos MySQL, el servidor de eventos.

El siguiente diagrama muestra los componentes de una implementación local completa de la plataforma AppDynamics APM. Muestra cómo interactúan los componentes para cumplir con la aplicación, la base de datos, la infraestructura, el monitoreo del usuario final y más.



A continuación, se explicará las partes básicas de una instalación de AppDynamics.

## 1.1 Controller Infrastructure

AppDynamics Controller es el servidor de administración central donde se almacenan y analizan todos los datos.

El Controller está desplegado en un servidor Glassfish y a su vez, usa MySQL como su base de datos predeterminada, donde almacena datos de configuración, datos de métricas, datos de snapshot de transacciones y eventos, y el historial de incidentes que ocurrieron (tanto los incidentes resueltos como los no resueltos se almacenan).

La versión del servidor Glassfish es Glassfish 4.1.1. y la última versión de AppDynamics incluye la versión 5.7.31 de MySQL.

## 1.2 Event Server

La base de datos MySQL integrada en el controlador almacena la métrica de la aplicación y los datos de configuración generados por el controlador. Si bien una base de datos integrada es suficiente para almacenar este tipo de datos, la naturaleza de alto volumen y rendimiento intensivo de los datos analíticos requiere un almacenamiento dedicado y escalable horizontalmente. En una implementación de AppDynamics, este rol lo desempeña el Event Server, el cual está desplegado en un Elasticsearch.

Puede implementar el Event Server en un solo nodo o en un clúster de tres o más nodos. Los clústeres son escalables horizontalmente, por lo que se pueden agregar nodos a medida que crecen sus requisitos de almacenamiento de datos. Un clúster también proporciona replicación y redundancia de datos, lo que ayuda a garantizar la integridad de los datos en caso de falla de un nodo.

## 1.3 App Agent Server

Los agentes de AppDynamics recopilan datos de sus servidores de aplicaciones y los envían al controlador. En AppDynamics existen agentes para JAVA, .NET, Node.js, PHP, Python.

## 1.4 Machine Agent

El Machine Agent es una aplicación Java autónoma que se puede ejecutar junto con los agentes de la aplicación o por separado. Esto significa que el monitoreo se puede extender a entornos fuera del ámbito de la aplicación que se monitorea. Se puede implementar en servidores de aplicaciones, servidores de bases de datos, servidores web, realmente cualquier cosa que ejecute Linux, UNIX, Windows o MAC. La funcionalidad proporcionada por el agente de máquina incluye:

- Informar métricas de hardware básicas del sistema operativo del servidor, por ejemplo, % de utilización de CPU y memoria, E / S de disco y red

- Informe de métricas transmitidas al controlador por extensiones
- Ejecución de scripts de corrección para acciones de políticas
- Ejecución de JVM Crash Guard

Se incluye una licencia de Machine Agent para cada máquina que tenga instalado un agente de aplicación con licencia. El Machine Agent recopila métricas de infraestructura de múltiples extensiones y las reenvía al controlador. También puede usar estas métricas para encontrar correlaciones entre los problemas de infraestructura en uno o más servidores y los problemas de rendimiento de las aplicaciones informados por los agentes de aplicaciones.

## 1.5 DB Agent

El DB Agent es un programa Java que recopila métricas de rendimiento sobre las instancias y los servidores de la base de datos tales como top de queries ejecutados, tiempo de la base de datos, total de ejecuciones, etc.

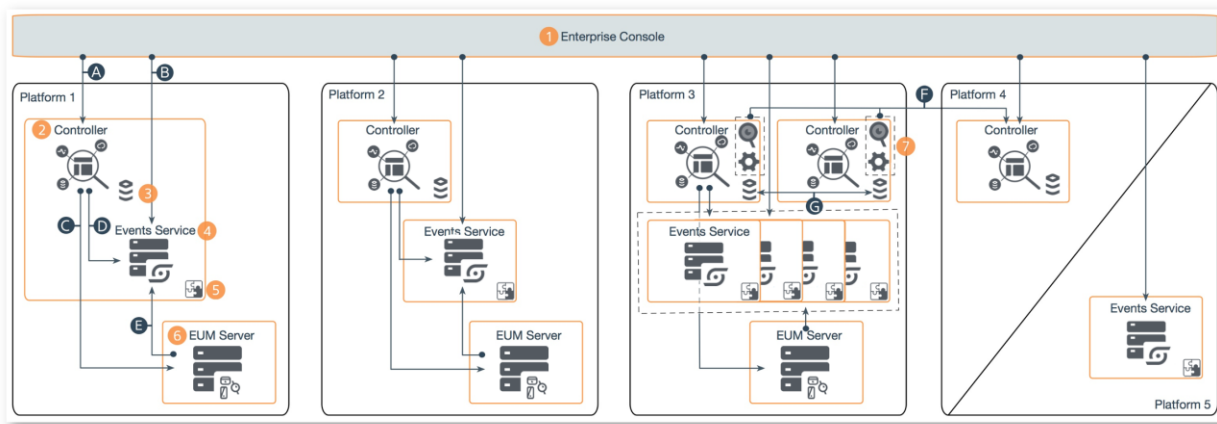
Para monitoreo de instancias se configura un collector de base de datos. Con estos collectores se puede monitorear SQL Server, Oracle, Sybase, MySQL, IBMDB2, MongoDB, PostgreSQL, AmazonRDS, CouchBase.

De forma predeterminada, muchas bases de datos y data store se detectan automáticamente cuando se realizan llamadas desde nodos equipados con App Agents, sin embargo, esta detección solo es básica.

## 2 Enterprise Console

Enterprise Console le permite instalar y administrar todo el ciclo de vida de las plataformas y componentes de AppDynamics locales nuevos o existentes. La aplicación proporciona una GUI y una interfaz de línea de comandos. Si el host de Enterprise Console deja de funcionar, no afecta a los controladores, el servicio de eventos ni los pares de alta disponibilidad (HA). Estos servicios seguirán ejecutándose independientemente de la aplicación. Luego, puede descubrir todas las plataformas en un nuevo host de Enterprise Console sin ningún impacto en los componentes.

El siguiente diagrama muestra cinco ejemplos de plataformas que pueden ser implementadas y administradas por Enterprise Console.



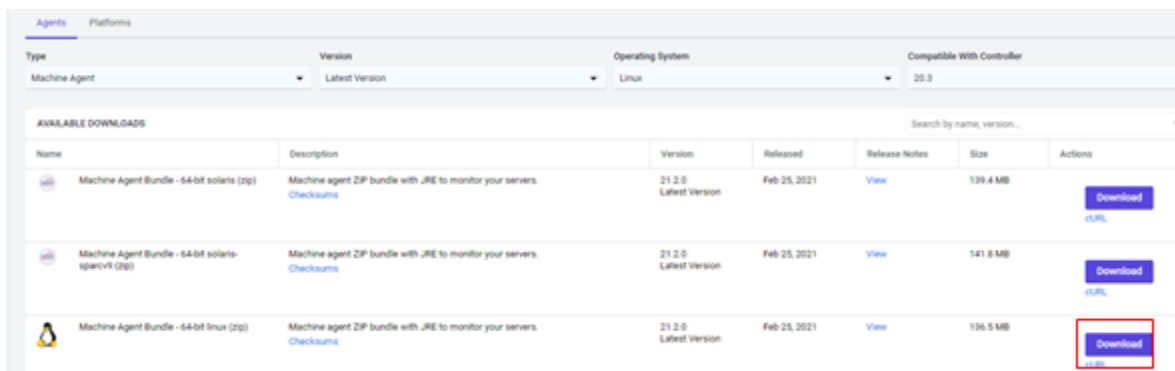
### 3 Machine Agent como servicio

Para la instalación de un Machine Agent como servicio en Linux con systemd se realizó el siguiente laboratorio:

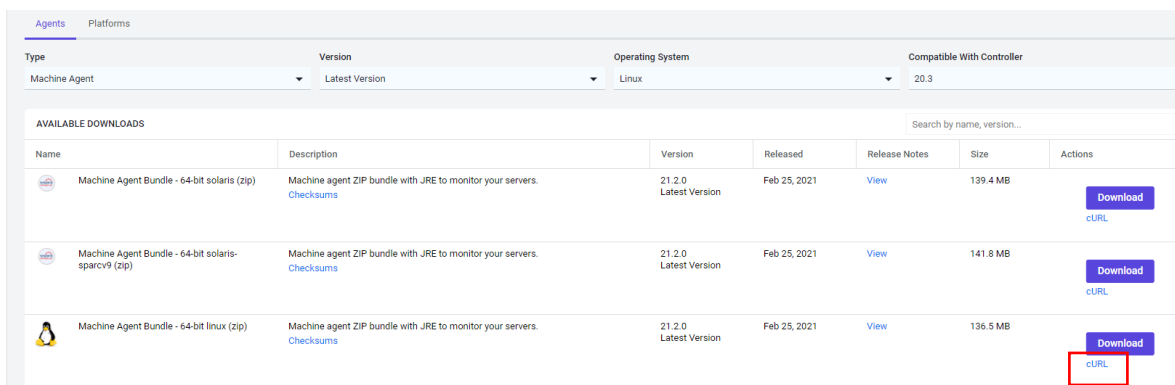
- Ambiente virtual.
- Sistema operativo: CentOS 7
  - CPU'S: 2
  - Memoria: 2 GB
  - Disco: 8 GB
- Versión del Controller: 20.3 Versión del agente: 21.2.0
- Link de descarga: <https://download.appdynamics.com/download/prox/download-file/machine-bundle/21.2.0.3052/machineagent-bundle-64bit-linux-21.2.0.3052.zip>

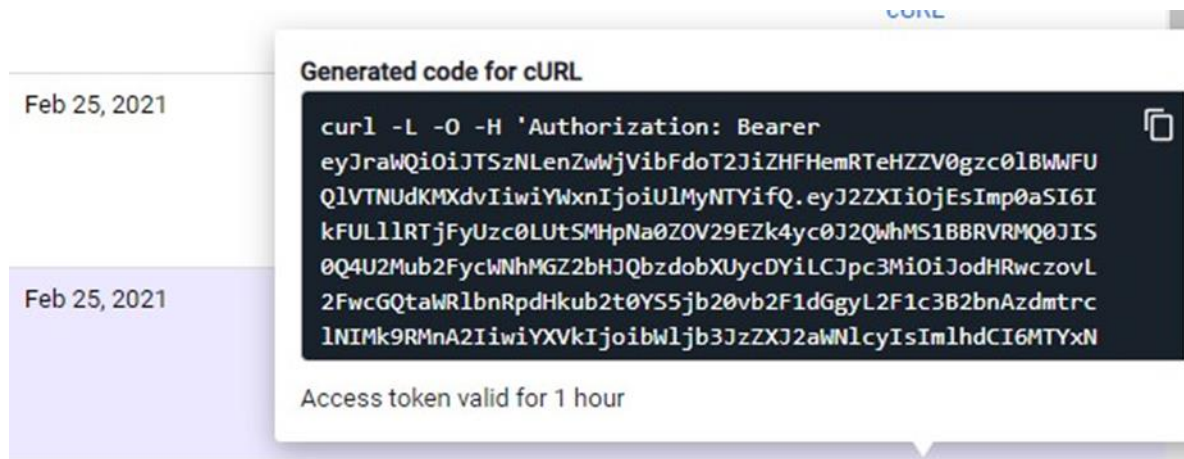
#### 3.1 Descarga del agente

Hay dos formas para descargar el agente en Linux, la primera es hacer una descarga directamente de la página de descargas de AppDynamics como se muestra a continuación.



La segunda manera es generando un token en la página de descargas en el apartado que dice cURL el cual generará un token que tendrá una caducidad de una hora.





Para este manual se utilizará el segundo método de instalación del machine-agent.

Antes de hacer la instalación verificamos que el sistema tenga el binario curl de la siguiente manera. Se usará el siguiente comando en la terminal **man curl**

```
[appdyn@localhost ~]$ man curl
```

Donde se desplegará el manual del binario curl

```
curl(1)                                Curl Manual                                curl(1)

NAME
    curl - transfer a URL

SYNOPSIS
    curl [options] [URL...]

DESCRIPTION
    curl is a tool to transfer data from or to a server, using one of the
```



Se copia y se pega el token que se generó en la página de las descargas

```
[appdyn@localhost ~]$ curl -L -O -H 'Authorization: Bearer eyJraWQiOiJTSzNLZWwW
jVibFdoT2JiZHFHemRteHZZV0gzcz01BWWFUQ1VTNUdKMxdvIiwiaWxnbG9kb3kiOiJ1bWNTYifQ.eyJ2ZXIiOiJ
EsImp0aSI6IkFUL1lRTjFyUzc0LUtSMHpNa0ZOV29EZk4yc0J2QWhMS1BBRVRMQ0JIS0Q4U2Mub2FycW
NhMGZ2bHJQbzdobXUycDYiLCJpc3MiOiJodHRwczovL2FwcGQtYWVlbnRpdHkub2t0YS5jb20vb2FldG
gyL2F1c3B2bnBnZmtrclNIMk9RMnA2IiwiaXVkbG9kb3kiOiJ1bWNTYifQ.eyJ2ZXIiOiJTSzNLZWwW
Q4OSwiZXBhIjoxNjE0OTU2ODg5L2F1c3B2bnBnZmtrclNIMk9RMnA2IiwiaXVkbG9kb3kiOiJ1bWNTYifQ.eyJ2ZXIiOiJTSzNLZWwW
MwdTBxZG5KVThWS2IwMnA3IiwiaXVkbG9kb3kiOiJ1bWNTYifQ.eyJ2ZXIiOiJTSzNLZWwW
QiXSwic3ViIjoicnFtYWdvc0B0ZWNUb2l1ZG1hLmNvbS5teCJ9.Mr-e98SiwXLOM8cNgFS2NVo6qgbc
EcJTLHCQyocBX9JmU79G-HbhkvarzZehnXFvT2-gTQ6h_XV83xRIu6-MVlhlk1CRvYaJuWlWw-y-0iyw
jb8f0zjYf93UMBScdYupirKdf5M_xM9k5D0EOcgVLdSiqQ1fEWa_0CLO_sAPTk2jYAYtTlKnI-9WvL
mwmxmRZBCwz5-ZZ5v9B0Irw6iBIMtnzRknxmqYQnct5mqMslAlvJ6p4BxMn1HaVuUgvveq5J4xl0Qk43
58gsruI8Avom9qgMGVi-FffAwhuSjl-3OyI099h8y7S9l0lIppv6XXNvn-Xz8FwRXU7zYe8atg;' 'ht
tps://download.appdynamics.com/download/prox/download-file/machine-bundle/21.2.0.
3052/machineagent-bundle-64bit-linux-21.2.0.3052.zip'
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
0	0	0	0	0	--:--:--	0:00:01	--:--:--
100	136M	100	136M	0	0	13.9M	0
					0:00:09	0:00:09	--:--:--
							16.7M

Para llevar un control de los agentes se crearán dos directorios con el binario mkdir los cuales se llamarán: appdynamics y dentro de este directorio se creará otro llamado machine-agent  
Ya descargado el archivo .zip se descomprimirá en el directorio machine-agent de la siguiente manera:

```
unzip machineagent-bundle-64bit-linux-21.2.0.3052.zip -d appdynamics/machine-agent/
```

### 3.2 Configuración del archivo Controller-info.xml

En la ruta machine-agent-home/conf/ se encontrará el archivo de configuración del agente donde se agregarán las configuraciones de agente con un editor de texto vi, vim o nano en las siguientes etiquetas.

```
<controller-host></controller-host>
<controller-port></controller-port>
<controller-ssl-enabled></controller-ssl-enabled>
<account-access-key></account-access-key>
<account-name></account-name>
<sim-enabled></sim-enabled>
```

Ya configurado el archivo controller-info.xml se modificara el archivo appdynamics-machine-agent.service que se encuentra en la ruta machine-agent-home/etc/systemd/system donde se modificarán las siguientes líneas:

```
Environment=MACHINE_AGENT_HOME=/machine-agent-home
Environment=JAVA_HOME=/machine-agent-home/jre
```

```
Type=simple

Environment=MACHINE_AGENT_HOME=/home/appdyn/appdynamics/machine-agent
Environment=JAVA_HOME=/home/appdyn/appdynamics/machine-agent/jre

# Specify agent system properties for systemd here by setting or editing JAVA_O
```

User=appdynamics-machine-agent

Environment=MACHINE\_AGENT\_USER=appdynamics-machine-agent

```
# Modify the next two lines to specify the user to run the machine agent as. Note that
# you will need to ensure that:
# 1. The controller-info.xml in the agent conf directory is readable by this user
# 2. The logs directory is writeable by this user
# 3. The scripts directory is writeable by this user
User=appdyn
Environment=MACHINE_AGENT_USER=appdyn
```

PermissionsStartOnly=true

```
# Run ExecStartPre with root-permissions
PermissionsStartOnly=true
```

Con permisos de root crear un link con el binario ln -s de  
/machine-agent-home/etc/systemd/system/appdynamics-machine-agent.service  
/etc/systemd/system/appdynamics-machine-agent.service

```
[root@localhost ~]# ln -s /home/appdyn/appdynamics/machine-agent/etc/systemd/system/appdynamics-m
achine-agent.service /etc/systemd/system/appdynamics-machine-agent.service
```

Se valida si se creó el *link* en la ruta /etc/systemd/system/

```
[appdyn@localhost system]$ ls -l
total 4
lrwxrwxrwx. 1 root root 91 Mar  4 04:34 appdynamics-machine-agent.service -> /home/appdyn/appdyn
amics/machine-agent/etc/systemd/system/appdynamics-machine-agent.service
```

Dar permisos 777 al directorio con el binario chmod de forma recursiva system en la ruta machine-agent-home/etc/systemd/system

chmod -Rf 777 system/

```
[appdyn@localhost system]$ cd ..
[appdyn@localhost systemd]$ chmod -Rf 777 system/
```

Con el usuario **root** ejecutar el binario `systemctl start appdynamics-machine-agent` iniciar el servicio

```
[root@localhost ~]# systemctl start appdynamics-machine-agent.service
```

Si todo está bien configurado no tendremos salida del binario anteriormente ejecutado

```
[root@localhost ~]# systemctl start appdynamics-machine-agent.service
[root@localhost ~]#
```

Usaremos el binario `systemctl` con el parámetro `status` para validar que funcione correctamente el servicio `systemctl status appdynamics-machine-agent`

```
[root@localhost ~]# systemctl status appdynamics-machine-agent.service
● appdynamics-machine-agent.service - AppDynamics Machine Agent
   Loaded: loaded (/home/appdyn/appdynamics/machine-agent/etc/systemd/system/appdynamics-machine-agent.service; linked; vendor preset: disabled)
   Active: active (running) since Thu 2021-03-04 05:37:29 CST; 1min 18s ago
     Process: 10636 ExecStartPre=/usr/bin/install -o $MACHINE_AGENT_USER -d $PIDDIR (code=exited, status=0/SUCCESS)
    Main PID: 10651 (java)
      CGroup: /system.slice/appdynamics-machine-agent.service
              └─10651 /home/appdyn/appdynamics/machine-agent/jre/bin/java -Xmx256m -Dlog4j.configu...

Mar 04 05:37:29 localhost.localdomain systemd[1]: Starting AppDynamics Machine Agent...
Mar 04 05:37:29 localhost.localdomain systemd[1]: Started AppDynamics Machine Agent.
Mar 04 05:37:29 localhost.localdomain sh[10638]: Using java executable at /home/appdyn/appdyn...va
Hint: Some lines were ellipsized, use -l to show in full.
```

Para dejar el servicio usaremos el mismo binario `systemctl` con el parámetro `enable` esto para que al iniciar el equipo se levante el agente

```
[root@localhost ~]# systemctl enable appdynamics-machine-agent.service
Created symlink from /etc/systemd/system/multi-user.target.wants/appdynamics-machine-agent.service to /home/appdyn/appdynamics/machine-agent/etc/systemd/system/appdynamics-machine-agent.service.
```

## 4 Agentes en Solaris para WebMethods

A continuación, se presenta el proceso de instalación de AppAgent y Machine Agent de Appdynamics en sistemas solaris para WebMethods.

1. Descargar los agentes de la página oficial.

Para poder descargarlos se debe de tener una cuenta registrada en appdynamics.com ya que si no se cuenta con un registro no se van a poder descargar los agentes.

- Para la descarga del Machine Agent se dirigirá a:

<https://download.appdynamics.com/download/prox/download-file/machine-bundle/20.4.0.2571/machineagent-bundle-64bit-solaris-sparcv9-20.4.0.2571.zip>

- Para la descarga del AppAgent de dirigirá a:

<https://download.appdynamics.com/download/prox/download-file/sun-jvm/20.4.0.29862/AppServerAgent-20.4.0.29862.zip>

Existe un archivo espacial para el monitoreo de volúmenes el cual va adjunto (FileSystemExtension).

2. Acceder al servidor en el cual se hará la instalación de los agentes

Crear tres directorios con el comando mkdir los cuales se llamarán appdynamics, AppAgent e instaladores.

Crear otro directorio de nombre machine-agent dentro del directorio appdynamics con el comando: mkdir machine-agent

3. Subir tres archivos a la carpeta de instaladores.

### Configuración Machineagent

- a. Posicionarse en la carpeta de instaladores y descomprimir el archivo zip con el siguiente comando. unzip Machineagent-XXX.zip -d ../appdynamics/machine-agent/
- b. Una vez descomprimido el machineagent mover el archivo FileSystemExtension al directorio de monitors que está en la ruta appdynamics/machine-agent/monitors con el comando mv appdynamics/machine-agent/monitors. Ya que el archivo esté en el lugar deseado descomprimir el archivo con el comando unzip FileSystemExtension

- c. Dar permisos al archivo filesystem-ext.sh con el comando `chmod 775 filesystem-ext.sh`
- d. Modificar el archivo `controller-info.xml` que está en la ruta /

`u01/apmusr/appdynamics/machine-agent/conf` con un editor de textos

- e. Posicionarse en la ruta `/u01/apmusr/appdynamics/machine-agent/monitors/analytics-agent`
- f. Modificar con un editor de texto el archivo `monitor.xml` en la etiqueta `<enabled>true</enabled>` guardar los cambios.
- g. Posicionarse en la ruta `/u01/apmusr/appdynamics/machine-agent/monitors/analytics-agent/conf` y modificar con un editor de texto el archivo `analytics-agent.properties` donde se cambiaran las variables:

```
ad.agent.name=analytics-(nombre del servidor)
ad.process.name=analytics-agent-(nombre del servidor)
ad.controller.url=http://10.184.48.226:8090
http.event.endpoint=http://10.184.48.227:9080
http.event.name=customer1
http.event.accountName=customer1_2fcad6d4-dc28-4ed0-be8a-9fd33c3a221a
http.event.accessKey=31e0e6da-2b04-41f5-9eb3-282a0fea5614
```

- h. Una vez abierto el archivo en el editor de textos cambiar los datos que están entre las etiquetas con los datos deseados

```
<controller-host>10.184.48.226</controller-host>
<controller-port>8090</controller-port>
<controller-ssl-enabled>false</controller-ssl-enabled>
<account-access-key>31e0e6da-2b04-41f5-9eb3-282a0fea5614</account-access-key>
<account-name>customer1</account-name>
<sim-enabled>true</sim-enabled>
```

### Configuración AppAgent

- a. Posicionarse en la carpeta de instaladores y descomprimir el archivo zip con el siguiente comando. `unzip AppAgentXXX.zip -d ../AppAgent/`
- b. Crear un enlace simbólico `javaagent` dentro del directorio `appdynamics` de la siguiente manera:

```
ln -s /u01/apmusr/AppAgent/ver20.*** /u01/apmusr/appdynamics/javaagent
```

donde los asteriscos serán el número de la versión del agente.

- c. Dar permisos a enlace javaagent con el comando `chmod 777 javaagent/`
- d. Posicionarse en la ruta `/u01/apmusr/appdynamics/javaagent/conf` y modificar el archivo `controller-info.xml` donde se modificarán las etiquetas siguientes.

```
<controller-host>10.184.48.226</controller-host>
<controller-port>8090</controller-port>
<controller-ssl-enabled>false</controller-ssl-enabled>
<application-name>Nombre que se indique</application-name>
<tier-name>nombre que se indique</tier-name>
<node-name>nombre del servidor</node-name>
<account-name>customer1</account-name>
<account-access-key>31e0e6da-2b04-41f5-9eb3-282a0fea5614</account-access-key>
```

### Configuración Wrapper

- a. Configurar wrapper. Si hay otra herramienta de monitoreo en la configuración de los wrappers es mandatorio que se comente la línea ya que si no se hace eso se tendrán issues.

- Agregar la línea:

```
wrapper.java.additional.600=-
javaagent:/u01/apmusr/appdynamics/javaagent/javaagent.jar
```

- Guardar los cambios
- Reiniciar el integration sever
- Iniciar machine agent

Una vez terminada la configuración del AppAgent y ver las transacciones de los servicios se levantará el machine agent con el siguiente comando posicionado en el directorio `machine-agent`

```
nohup /u01/apmusr/appdynamics/machine-agent/jre/bin/java -Xmx512m -
Dlog4j.configuration=file:/u01/apmusr/appdynamics/machine-
agent/conf/logging/log4j.xml -jar /u01/apmusr/appdynamics/machine-
agent/machineagent.jar &
```

## 5 Alertas configuradas

En AppDynamics se pueden configurar Health Rules de salud para la disponibilidad de agentes, las cuales serán de ayuda en caso de que un servidor este apagado. En todas las aplicaciones monitoreadas se encuentra una política como la siguiente:

Policies

Corresponsalias

Create







Edit

Delete

Copy

Enable

Disable

	Name ↑	Trigger	Enabled
	CORRESPONSALIAS	Health Rule Violation Events: Health Rule Violation Started - Critical, Health Rule Violati...  Fire when events occur on ANY objects	
	Corresponsalias AppAgent Apagado	Health Rule Violation Events: Health Rule Violation Started - Critical, Health Rule Violati...  Fire when events occur on ANY objects	
	TDC y Remesas	Health Rule Violation Events: Health Rule Violation Started - Critical, Health Rule Violati...  Fire when events occur on ANY objects	

Esta política se activará en cuanto una alerta critica inicie y finalice.

Edit Policy - Corresponsalias AppAgent Apagado

Trigger    **Health Rule Scope**    Object Scope    Actions

Name

Enabled ☒

Execute actions in batch ☒

**This Policy will fire when any of these Events occur**

Health Rule Violation Events

☐ Health Rule Violation Started - Warning  
☒ Health Rule Violation Started - Critical  
☐ Health Rule Violation Continues - Warning  
☐ Health Rule Violation Continues - Critical  
☐ Health Rule Violation Upgraded - Warning to Critical  
☐ Health Rule Violation Downgraded - Critical to Warning  
☐ Health Rule Violation Ended - Warning  
☒ Health Rule Violation Ended - Critical  
☐ Health Rule Violation Canceled - Warning  
☐ Health Rule Violation Canceled - Critical

Other Events

☐ Slow Transactions  
☐ Code Problems  
☐ Application Changes  
☐ Server Crashes  
☐ AppDynamics Config Warnings  
☐ Discovery  
☐ Synthetic Availability  
☐ Synthetic Performance  
☐ Mobile Crash  
☐ Errors

Custom Events ?

+    ✎    🗑

Type	Properties
No Custom Events Selected	

Cancel    Save

La regla de salud ligada a esta política es:

Edit Policy - Corresponsalias AppAgent Apagado

Trigger

Health Rule Scope

Object Scope

Actions

Health Rules that trigger this Policy:

Any Health Rule

These Health Rules

Name	Type	Status
<div><div></div>App Agent Apagado</div>	Node Health - Hardware, JVM, CLR (cpu, heap, disk I/O, etc)	<div></div>

Finalmente, cuando la política se active las acciones que se ejecutarán son el envío de correo a los correos indicados.

Edit Policy - Corresponsalias AppAgent Apagado

Trigger

Health Rule Scope

Object Scope

Actions

Actions to Execute

Name	Type
<div><div></div>cmaxxo.mloxxo@oxxo.com,monitoreooper@oxxo.com</div>	Email
<div><div></div>rqmagos@tecnomedia.com.mx,jlizepeda@tecnomedia.com.m:</div>	Email
<div><div></div>usufcsoportel3wm@oxxo.com</div>	Email



## 6 Referencias

- <https://docs.appdynamics.com/display/PRO45/AppDynamics+Application+Performance+Monitoring+Platform>
- <https://docs.appdynamics.com/display/PRO45/Controller+Component+Versions>
- <https://docs.appdynamics.com/display/PRO45/Events+Service+Deployment#:~:text=The%20AppDynamics%20Events%20Service%20is,for%20on%2Dpremises%20deployments%20only.>
- <https://docs.appdynamics.com/display/PRO45/Install+App+Server+Agents>
- <https://docs.appdynamics.com/display/PRO45/Standalone+Machine+Agent>
- <https://docs.appdynamics.com/display/PRO45/JVM+Crash+Guard>
- <https://docs.appdynamics.com/display/PRO45/Monitor+Databases>
- <https://docs.appdynamics.com/display/PRO45/Enterprise+Console>
- <https://docs.appdynamics.com/display/PRO45/Enterprise+Console+Command+Line>
- <https://docs.appdynamics.com/display/PRO45/Upgrade+the+Enterprise+Console>