

AppDynamics OnPrem Controller Installation & Security Guide

Table of Contents

1. [Prerequisites](#)
2. [Database Setup](#)
3. [Controller Installation](#)
4. [Initial Configuration](#)
5. [SSL/TLS Certificate Configuration](#)
6. [Load Balancer Configuration](#)
7. [High Availability Setup](#)
8. [Security Hardening](#)
9. [Performance Tuning](#)
10. [Post-Installation Configuration](#)
11. [Verification and Testing](#)

Prerequisites

System Requirements

- **Operating System:** Linux (RHEL/CentOS 7+, Ubuntu 18.04+) or Windows Server 2016+
- **RAM:** Minimum 8GB, Recommended 16GB+ for production
- **CPU:** 4+ cores for production environments
- **Disk Space:** 100GB+ for controller, additional space for data retention
- **Network:** Ports 8090 (HTTP), 8181 (HTTPS), 9300-9400 (agent communication)

Database Requirements

- **Supported Databases:** MySQL 5.7+, Oracle 12c+, SQL Server 2016+
- **Database Server:** Separate dedicated server recommended for production
- **Storage:** Fast SSD storage with adequate IOPS

Load Balancer Requirements

- **Supported:** F5, HAProxy, NGINX, AWS ALB, Azure Load Balancer
- **Features:** SSL termination, session persistence, health checks

Database Setup

MySQL Configuration

```
sql

-- Create AppDynamics database and user
CREATE DATABASE controller CHARACTER SET utf8 COLLATE utf8_bin;
CREATE USER 'appdynamics'@'%' IDENTIFIED BY 'SecurePassword123!';
GRANT ALL PRIVILEGES ON controller.* TO 'appdynamics'@'%';
FLUSH PRIVILEGES;
```

MySQL Configuration File (/etc/mysql/mysql.conf.d/mysqld.cnf)

```
ini

[mysqld]
# Basic Settings
datadir = /var/lib/mysql
socket = /var/run/mysqld/mysqld.sock
pid-file = /var/run/mysqld/mysqld.pid

# AppDynamics Specific Settings
max_connections = 500
innodb_buffer_pool_size = 4G
innodb_log_file_size = 256M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 120
query_cache_size = 0
query_cache_type = 0

# Character Set
character-set-server = utf8
collation-server = utf8_bin

# Security
bind-address = 0.0.0.0
ssl-ca = /etc/mysql/ssl/ca.pem
ssl-cert = /etc/mysql/ssl/server-cert.pem
ssl-key = /etc/mysql/ssl/server-key.pem
```

Controller Installation

Download and Extract

```
bash

# Download AppDynamics Controller
wget https://download.appdynamics.com/download/controller/4.5.19/appdynamics-controller-4.5.19.0.tar.gz

# Extract to installation directory
sudo mkdir -p /opt/appdynamics
cd /opt/appdynamics
sudo tar -xzf appdynamics-controller-4.5.19.0.tar.gz
sudo chown -R appdynamics:appdynamics /opt/appdynamics
```

Create AppDynamics User

```
bash

# Create dedicated user for AppDynamics
sudo useradd -r -m -d /home/appdynamics -s /bin/bash appdynamics
sudo mkdir -p /home/appdynamics/.ssh
sudo chown -R appdynamics:appdynamics /home/appdynamics
```

Configure Response File

Create `/opt/appdynamics/response.varfile`:

```
properties
```

Server Configuration

serverHostName=appdynamics-controller.company.com

sys.languageId=en

sys.installationDir=/opt/appdynamics/controller

Database Configuration

dbHostName=mysql-server.company.com

dbPort=3306

dbUserName=appdynamics

dbPassword=SecurePassword123!

dbSchemaName=controller

dbUseSSL=true

Root User Configuration

rootUserPassword=AdminPassword123!

mysqlRootPassword=MySQLRootPassword123!

License Configuration

licenseFile=/opt/appdynamics/license.lic

Run Installation

bash

`sudo -u appdynamics /opt/appdynamics/platform-admin/platform-admin.sh install-controller --response-file /opt/app`

Initial Configuration

Controller Configuration File

Edit `/opt/appdynamics/controller/bin/controller.sh`:

bash

JVM Configuration

JAVA_OPTS="\$JAVA_OPTS -Xms4g -Xmx8g"

JAVA_OPTS="\$JAVA_OPTS -XX:+UseG1GC"

JAVA_OPTS="\$JAVA_OPTS -XX:MaxGCPauseMillis=200"

JAVA_OPTS="\$JAVA_OPTS -Dappdynamics.controller.rootUser.username=root@system"

Network Configuration

JAVA_OPTS="\$JAVA_OPTS -Dappdynamics.controller.hostName=appdynamics-controller.company.com"

JAVA_OPTS="\$JAVA_OPTS -Dappdynamics.controller.port=8090"

JAVA_OPTS="\$JAVA_OPTS -Dappdynamics.controller.ssl.port=8181"

Database Connection Pool Configuration

Create `/opt/appdynamics/controller/db/db-config.xml`:

xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<database-configurations>
```

```
  <database-configuration name="controller">
```

```
    <connection-pool-configuration>
```

```
      <max-pool-size>50</max-pool-size>
```

```
      <min-pool-size>10</min-pool-size>
```

```
      <initial-pool-size>10</initial-pool-size>
```

```
      <connection-timeout>30000</connection-timeout>
```

```
      <idle-timeout>600000</idle-timeout>
```

```
      <max-lifetime>1800000</max-lifetime>
```

```
    </connection-pool-configuration>
```

```
  </database-configuration>
```

```
</database-configurations>
```

SSL/TLS Certificate Configuration

Generate or Import Custom Certificates

bash

```
# Create certificate directory
sudo mkdir -p /opt/appdynamics/controller/ssl
cd /opt/appdynamics/controller/ssl

# Import your custom certificate and private key
sudo cp /path/to/your/certificate.crt ./controller.crt
sudo cp /path/to/your/private.key ./controller.key
sudo cp /path/to/your/ca-bundle.crt ./ca-bundle.crt

# Create keystore from certificate
sudo openssl pkcs12 -export -in controller.crt -inkey controller.key -out controller.p12 -name controller -password pass

# Import into Java keystore
sudo keytool -importkeystore -deststorepass keystorepassword -destkeypass keystorepassword -destkeystore controller

# Import CA certificate
sudo keytool -import -alias ca-cert -file ca-bundle.crt -keystore controller.jks -storepass keystorepassword -noprompt

sudo chown -R appdynamics:appdynamics /opt/appdynamics/controller/ssl
```

Configure SSL in Controller

Edit `/opt/appdynamics/controller/bin/controller.sh`:

```
bash

# SSL Configuration
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=/opt/appdynamics/controller/ssl/controller.jks"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=keystorepassword"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=/opt/appdynamics/controller/ssl/controller.jks"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=keystorepassword"
JAVA_OPTS="$JAVA_OPTS -Dappdynamics.controller.ssl.enabled=true"
```

Load Balancer Configuration

HAProxy Configuration (`/etc/haproxy/haproxy.cfg`)

```
haproxy
```

global

```
log stdout local0
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin
stats timeout 30s
user haproxy
group haproxy
daemon
```

defaults

```
mode http
log global
option httplog
option dontlognull
option redispatch
retries 3
timeout http-request 10s
timeout queue 1m
timeout connect 10s
timeout client 1m
timeout server 1m
timeout http-keep-alive 10s
timeout check 10s
```

Frontend for HTTPS

frontend appdynamics_frontend

```
bind *:443 ssl crt /etc/ssl/certs/appdynamics.pem
bind *:80
redirect scheme https if !{ ssl_fc }
```

Security headers

```
http-response set-header Strict-Transport-Security "max-age=31536000; includeSubDomains"
http-response set-header X-Frame-Options "SAMEORIGIN"
http-response set-header X-Content-Type-Options "nosniff"
```

Route to appropriate backend

```
acl is_controller hdr(host) -i appdynamics-controller.company.com
use_backend appdynamics_controllers if is_controller
```

Backend for Controllers

backend appdynamics_controllers

```
balance roundrobin
option httpchk GET /controller/rest/serverstatus
```

cookie JSESSIONID prefix nocache

server controller1 10.0.1.10:8181 check ssl verify none cookie controller1

server controller2 10.0.1.11:8181 check ssl verify none cookie controller2

Stats page

listen stats

bind *:8404

stats enable

stats uri /stats

stats refresh 30s

stats admin if TRUE

NGINX Configuration (/etc/nginx/sites-available/appdynamics)

nginx


```
upstream appdynamics_controllers {
    least_conn;
    server 10.0.1.10:8181 max_fails=3 fail_timeout=30s;
    server 10.0.1.11:8181 max_fails=3 fail_timeout=30s;
}

server {
    listen 80;
    server_name appdynamics-controller.company.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name appdynamics-controller.company.com;

    # SSL Configuration
    ssl_certificate /etc/ssl/certs/appdynamics.crt;
    ssl_certificate_key /etc/ssl/private/appdynamics.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    # Security Headers
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;

    # Proxy Configuration
    location / {
        proxy_pass https://appdynamics_controllers;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # WebSocket support
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

```

# Timeouts
proxy_connect_timeout 60s;
proxy_send_timeout 60s;
proxy_read_timeout 60s;
}

# Health check endpoint
location /controller/rest/serverstatus {
    proxy_pass https://appdynamics_controllers;
    access_log off;
}
}

```

High Availability Setup

Controller Clustering Configuration

Edit `/opt/appdynamics/controller/bin/controller.sh` on each node:

```

bash

# Cluster Configuration
JAVA_OPTS="$JAVA_OPTS -Dappdynamics.controller.cluster.enabled=true"
JAVA_OPTS="$JAVA_OPTS -Dappdynamics.controller.cluster.node.id=controller1"
JAVA_OPTS="$JAVA_OPTS -Dappdynamics.controller.cluster.members=controller1:10.0.1.10,controller2:10.0.1.11"
JAVA_OPTS="$JAVA_OPTS -Dappdynamics.controller.cluster.data.dir=/opt/appdynamics/controller/data/cluster"

```

Shared Storage Configuration

```

bash

# Create shared storage mount points
sudo mkdir -p /opt/appdynamics/shared/{data,logs,backups}

# Mount NFS shares (add to /etc/fstab)
nfs-server.company.com:/exports/appdynamics/data /opt/appdynamics/shared/data nfs defaults 0 0
nfs-server.company.com:/exports/appdynamics/logs /opt/appdynamics/shared/logs nfs defaults 0 0
nfs-server.company.com:/exports/appdynamics/backups /opt/appdynamics/shared/backups nfs defaults 0 0

```

Security Hardening

Firewall Configuration (UFW)

```
bash
```

```
# Reset and set defaults
```

```
sudo ufw --force reset
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
# Allow SSH
```

```
sudo ufw allow 22/tcp
```

```
# Allow load balancer access only
```

```
sudo ufw allow from 10.0.2.0/24 to any port 8181 proto tcp comment "Load Balancer to Controller HTTPS"
```

```
sudo ufw allow from 10.0.2.0/24 to any port 8090 proto tcp comment "Load Balancer to Controller HTTP"
```

```
# Allow agent communication
```

```
sudo ufw allow 9300:9400/tcp comment "Agent Communication"
```

```
# Allow database access (if local)
```

```
sudo ufw allow from 10.0.1.0/24 to any port 3306 proto tcp comment "Database Access"
```

```
# Enable firewall
```

```
sudo ufw enable
```

System Security Configuration

```
bash
```

```
# Disable unnecessary services
```

```
sudo systemctl disable cups
```

```
sudo systemctl disable avahi-daemon
```

```
sudo systemctl disable bluetooth
```

```
# Configure secure SSH
```

```
sudo sed -i 's/#PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
```

```
sudo sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
```

```
sudo systemctl restart sshd
```

```
# Set file permissions
```

```
sudo chmod 600 /opt/appdynamics/controller/ssl/*
```

```
sudo chmod 755 /opt/appdynamics/controller/bin/controller.sh
```

```
sudo chown -R appdynamics:appdynamics /opt/appdynamics
```

Application Security Configuration

Create `/opt/appdynamics/controller/conf/security.properties`:

properties

Security Configuration

security.authentication.method=LDAP

security.session.timeout=3600

security.password.policy.enabled=true

security.password.policy.minLength=12

security.password.policy.requireUppercase=true

security.password.policy.requireLowercase=true

security.password.policy.requireNumbers=true

security.password.policy.requireSpecialChars=true

LDAP Configuration

ldap.server.url=ldaps://ldap.company.com:636

ldap.server.bindDN=cn=appdynamics,ou=service,dc=company,dc=com

ldap.server.bindPassword=LDAPPassword123!

ldap.user.searchBase=ou=users,dc=company,dc=com

ldap.user.searchFilter=(uid={0})

ldap.group.searchBase=ou=groups,dc=company,dc=com

ldap.group.searchFilter=(member={0})

Performance Tuning

JVM Tuning

Edit `/opt/appdynamics/controller/bin/controller.sh`:

bash

Memory Configuration

```
JAVA_OPTS="$JAVA_OPTS -Xms8g -Xmx8g"  
JAVA_OPTS="$JAVA_OPTS -XX:NewRatio=2"  
JAVA_OPTS="$JAVA_OPTS -XX:MetaspaceSize=512m"  
JAVA_OPTS="$JAVA_OPTS -XX:MaxMetaspaceSize=1g"
```

Garbage Collection

```
JAVA_OPTS="$JAVA_OPTS -XX:+UseG1GC"  
JAVA_OPTS="$JAVA_OPTS -XX:MaxGCPauseMillis=200"  
JAVA_OPTS="$JAVA_OPTS -XX:G1HeapRegionSize=16m"  
JAVA_OPTS="$JAVA_OPTS -XX:G1ReservePercent=10"  
JAVA_OPTS="$JAVA_OPTS -XX:G1MixedGCCountTarget=8"
```

GC Logging

```
JAVA_OPTS="$JAVA_OPTS -Xloggc:/opt/appdynamics/controller/logs/gc.log"  
JAVA_OPTS="$JAVA_OPTS -XX:+UseGCLogFileRotation"  
JAVA_OPTS="$JAVA_OPTS -XX:NumberOfGCLogFiles=5"  
JAVA_OPTS="$JAVA_OPTS -XX:GCLogFileSize=10M"
```

Performance Monitoring

```
JAVA_OPTS="$JAVA_OPTS -XX:+UnlockDiagnosticVMOptions"  
JAVA_OPTS="$JAVA_OPTS -XX:+LogVMOutput"
```

System Performance Tuning

bash

Kernel parameters for high performance

```
echo 'net.core.somaxconn = 65535' | sudo tee -a /etc/sysctl.conf  
echo 'net.core.netdev_max_backlog = 5000' | sudo tee -a /etc/sysctl.conf  
echo 'net.ipv4.tcp_max_syn_backlog = 65535' | sudo tee -a /etc/sysctl.conf  
echo 'net.ipv4.tcp_fin_timeout = 30' | sudo tee -a /etc/sysctl.conf  
echo 'net.ipv4.tcp_keepalive_time = 1200' | sudo tee -a /etc/sysctl.conf  
echo 'net.ipv4.tcp_max_tw_buckets = 400000' | sudo tee -a /etc/sysctl.conf
```

File descriptor limits

```
echo 'appdynamics soft nofile 65536' | sudo tee -a /etc/security/limits.conf  
echo 'appdynamics hard nofile 65536' | sudo tee -a /etc/security/limits.conf  
echo 'appdynamics soft nproc 32768' | sudo tee -a /etc/security/limits.conf  
echo 'appdynamics hard nproc 32768' | sudo tee -a /etc/security/limits.conf
```

```
sudo sysctl -p
```

Post-Installation Configuration

Systemd Service Configuration

Create `/etc/systemd/system/appdynamics-controller.service`:

```
ini

[Unit]
Description=AppDynamics Controller
After=network.target mysql.service

[Service]
Type=forking
User=appdynamics
Group=appdynamics
Environment=JAVA_HOME=/usr/lib/jvm/java-8-oracle
ExecStart=/opt/appdynamics/controller/bin/controller.sh start
ExecStop=/opt/appdynamics/controller/bin/controller.sh stop
ExecReload=/opt/appdynamics/controller/bin/controller.sh restart
PIDFile=/opt/appdynamics/controller/bin/controller.pid
TimeoutStartSec=300
TimeoutStopSec=300
Restart=on-failure
RestartSec=30

[Install]
WantedBy=multi-user.target
```

Enable and start the service:

```
bash

sudo systemctl daemon-reload
sudo systemctl enable appdynamics-controller
sudo systemctl start appdynamics-controller
```

Data Retention Configuration

Configure data retention policies in the Controller UI or via REST API:

```
bash
```

Example: Set metric data retention to 90 days

```
curl -X POST -H "Content-Type: application/json" \  
-u "root@system:AdminPassword123!" \  
"https://appdynamics-controller.company.com:8181/controller/rest/configuration/retention" \  
-d '{  
  "metricDataRetentionPeriod": 90,  
  "eventDataRetentionPeriod": 365,  
  "snapshotRetentionPeriod": 30  
'
```

Monitoring and Alerting Setup

bash

Create monitoring scripts

```
sudo mkdir -p /opt/appdynamics/monitoring
```

Controller health check script

```
cat > /opt/appdynamics/monitoring/health_check.sh << 'EOF'
```

```
#!/bin/bash
```

```
CONTROLLER_URL="https://appdynamics-controller.company.com:8181"
```

```
STATUS=$(curl -s -k -o /dev/null -w "%{http_code}" "$CONTROLLER_URL/controller/rest/serverstatus")
```

```
if [ "$STATUS" = "200" ]; then
```

```
  echo "Controller is healthy"
```

```
  exit 0
```

```
else
```

```
  echo "Controller health check failed with status: $STATUS"
```

```
  exit 1
```

```
fi
```

```
EOF
```

```
chmod +x /opt/appdynamics/monitoring/health_check.sh
```

Verification and Testing

Initial Verification Steps

bash

Check service status

```
sudo systemctl status appdynamics-controller
```

Verify network connectivity

```
netstat -tlnp | grep :8181
```

```
netstat -tlnp | grep :8090
```

Check logs

```
tail -f /opt/appdynamics/controller/logs/server.log
```

Test SSL certificate

```
openssl s_client -connect appdynamics-controller.company.com:8181 -servername appdynamics-controller.company.com
```

Verify database connectivity

```
mysql -h mysql-server.company.com -u appdynamics -p controller -e "SELECT COUNT(*) FROM application;"
```

Load Balancer Testing

bash

Test load balancer health checks

```
curl -k https://your-load-balancer.company.com/controller/rest/serverstatus
```

Test SSL termination

```
curl -I https://your-load-balancer.company.com/controller
```

Verify session persistence

```
for i in {1..5}; do
```

```
    curl -c cookies.txt -b cookies.txt -k https://your-load-balancer.company.com/controller/auth
```

```
done
```

Security Testing

bash

SSL/TLS verification

```
nmap --script ssl-enum-ciphers -p 443 your-load-balancer.company.com
```

Port scanning verification

```
nmap -sS your-controller-server.company.com
```

Certificate verification

```
openssl s_client -connect your-load-balancer.company.com:443 -servername your-load-balancer.company.com
```

Performance Testing

bash

Memory usage monitoring

```
ps aux | grep controller
```

```
free -h
```

Disk I/O monitoring

```
iostat -x 1
```

Network monitoring

```
iftop -P -n -N
```

Database performance

```
SHOW PROCESSLIST;
```

```
SHOW ENGINE INNODB STATUS;
```

Maintenance and Backup

Automated Backup Script

bash

```
#!/bin/bash
# /opt/appdynamics/scripts/backup.sh

BACKUP_DIR="/opt/appdynamics/shared/backups"
DATE=$(date +%Y%m%d_%H%M%S)
RETENTION_DAYS=30

# Stop controller
sudo systemctl stop appdynamics-controller

# Database backup
mysqldump -h mysql-server.company.com -u appdynamics -p'SecurePassword123!' \
--single-transaction --routines --triggers controller > \
"$BACKUP_DIR/controller_db_$DATE.sql"

# Configuration backup
tar -czf "$BACKUP_DIR/controller_config_$DATE.tar.gz" \
/opt/appdynamics/controller/conf \
/opt/appdynamics/controller/ssl

# Start controller
sudo systemctl start appdynamics-controller

# Cleanup old backups
find "$BACKUP_DIR" -name "*.sql" -mtime +$RETENTION_DAYS -delete
find "$BACKUP_DIR" -name "*.tar.gz" -mtime +$RETENTION_DAYS -delete

echo "Backup completed: $DATE"
```

This comprehensive guide covers the complete installation, configuration, and security setup of AppDynamics OnPrem controllers with load balancers and custom certificates. Remember to adapt specific configurations to your environment and security requirements.