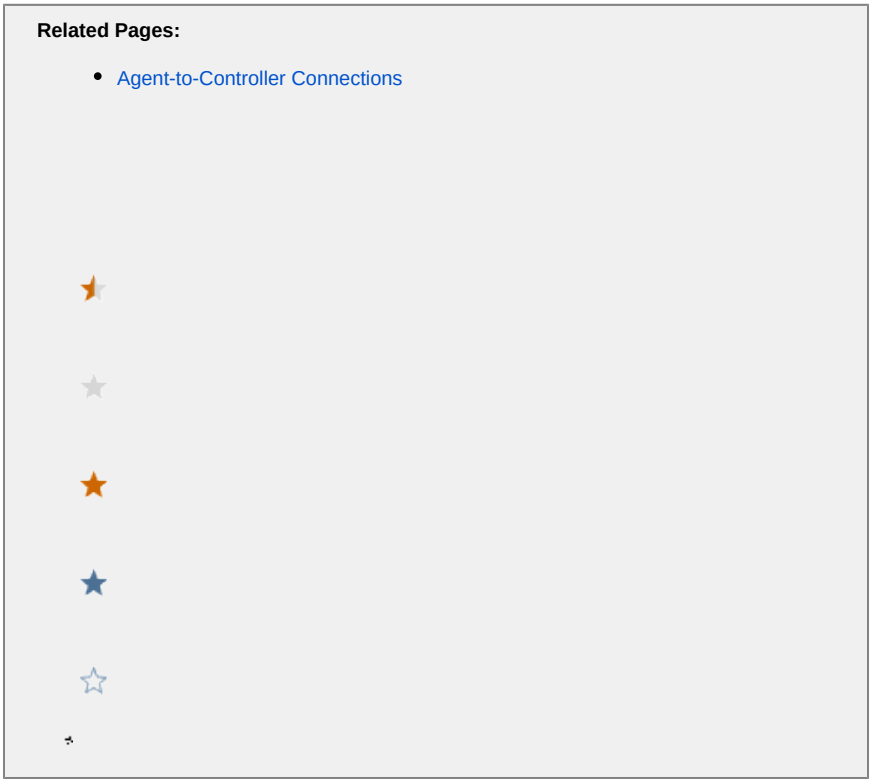
Set the Security Protocol



This page describes the security protocol used by an on-premises Controller, and how you can modify it.

Default Security Protocol

The Controller secures connections using TLSv1.2 by default. However, you can change the security protocols used by the Controller if needed. For instance, you need to change the protocol if you are using agents that don't support TLSv1.2. These agents include:

- Java Agent version 3.8.1 or earlier (see Agent and Controller Compatibility for complete SSL compatibility information)
- .NET Agent running on .NET Framework 4.5 or earlier

If upgrading the agents or .NET framework is not possible, you will need to enable TLSv1 and SSL3 on the Controller using the `asadmin` comm and-line utility. To use the utility, you will need to supply the password configured for the root user for the Controller.

These changes require a restart of the Controller application server, which results in a brief service downtime. You may wish to apply these change when the downtime will have the least impact.

To maintain a secure environment, APIs that are downstream of the Controller should also use TLS. If SSL3 is required, you can enable it. See the Oracle JDK 8 documentation.

Enable TLS for a Controller

1. Open a browser and navigate to the Enterprise Console GUI:

```
http(s)://<hostname>:<port>
```

9191 is the default port.
2. Navigate to AppServer Configurations by choosing the platform, **Configurations**, **Controller Settings**, and **Appserver Configurations**.
3. In the Domain Protocols box on the JVM Options tab, edit the `configs.config.server-config.network-config.protocols.protocol.http-listener-2.ssl.tls-enabled=false` parameter to true.
4. Click **Save**.

You do not need to restart the Controller application server since the configuration change job automatically does so for you.

Enabling Stronger Encryption Keys

By default, the Controller's embedded Java runtime only supports up to 128-bit encryption key lengths for secure connections. You can, however, enable up to 256-bit encryption keys so the Controller can establish connections using the stronger ciphers.

To enable stronger keys in encryption keys in the Controller, follow the instructions for the Controller version you are running.

The Controller versions 4.5.4 and higher are bundled with JDK 8u181, so you no longer need to download the Unlimited Strength Jurisdiction Policy Files from Oracle.

To enable unlimited cryptography:

1. Set the new Security property `crypto.policy` to `'unlimited'` in the `<JRE>/lib/security/java.security` file.

   By default, the property `crypto.policy` is undefined. If the property is undefined and the legacy JCE jurisdiction files don't exist in the legacy lib/security directory, then the default cryptographic level will remain at 'limited' (128-bit encryption).

2. Restart the Controller app server.

In older releases of JDK, the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files files have to be downloaded and installed separately to allow unlimited cryptography to be used by the JDK.

1. Download the Unlimited Strength Jurisdiction Policy Files from the following location: http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
2. Install the policy files in the JRE installed under the Enterprise Console "Platform Installation Path".
3. Restart the Controller app server.

After restarting the Controller app server, the following cipher suites become available:

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_DSS_WITH_AES_256_CBC_SHA