If you have a lot of Redis databases or you are using RedisInsight as part of some automated workflow, you might want to add databases programmatically.

Now this is possible using our experimental REST API. Below is the documentation for the endpoints required to add databases.

Note that this API should not be considered stable at this point and might change or break entirely in future releases. Do not rely on this API for production.

**Add Redis database**

Used to add Redis databases to RedisInsight.

**URL**: /api/instance/

**Method**: POST

**Body Type**: JSON

**Auth required**: NO

### Parameters

These are the required parameters for any type of database.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>A nick name for the Redis database. Any string is valid</td>
</tr>
<tr>
<td>connectionType</td>
<td>string</td>
<td>One of &quot;STANDALONE&quot;, &quot;CLUSTER&quot; or &quot;SENTINEL&quot;. For any Redis Enterprise database (even with database clustering enabled), use &quot;STANDALONE&quot;</td>
</tr>
</tbody>
</table>

The remaining parameters depend on the connection type.

**Standalone database parameters**

Standalone databases are added using `connectionType`: "STANDALONE".

The following additional parameters are required for standalone databases.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>string</td>
<td>The hostname or IP address of your Redis database</td>
</tr>
<tr>
<td>port</td>
<td>number</td>
<td>The port your Redis database is listening on. It should be an integer. (optional) The password for your Redis database.</td>
</tr>
</tbody>
</table>
TLS parameters for the database

**Parameter** | **Type** | **Description**
---|---|---
tls | object | (optional) TLS parameters for the database

Example

```json
{
   "name": "QA Redis DB",
   "connectionType": "STANDALONE",
   "host": "redis.acme.com",
   "port": 6379
}
```

Redis cluster database parameters

Redis Cluster databases are added using `connectionType: "CLUSTER"`. The following additional parameters are required for Redis Cluster databases.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seedNodes</td>
<td>array</td>
<td>An array of objects describing the nodes of the cluster. At least one node should be specified. The objects must contain properties <code>host</code> (string) and <code>port</code> (integer)</td>
</tr>
<tr>
<td>password</td>
<td>string</td>
<td>(optional) The password for your Redis database.</td>
</tr>
<tr>
<td>tls</td>
<td>object</td>
<td>(optional) TLS parameters for the database</td>
</tr>
</tbody>
</table>

Example

```json
{
   "name": "QA Redis Cluster DB",
   "connectionType": "CLUSTER",
   "seedNodes": [
      {
         "host": "redis-cluster-node-1.acme.com",
         "port": 6379
      }
   ]
}
```

Sentinel-monitored database parameters

Sentinel-monitored databases are added using `connectionType: "SENTINEL"`. The following additional parameters are required for standalone databases.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentinelHost</td>
<td>string</td>
<td>The hostname or IP address of one of the sentinel instances</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sentinelPort</td>
<td>number</td>
<td>The hostname or IP address of one of the sentinel instances</td>
</tr>
<tr>
<td>sentinelPassword</td>
<td>string</td>
<td>(optional) The password for the sentinel instances</td>
</tr>
<tr>
<td>sentinelMaster</td>
<td>object</td>
<td>Information about the monitored database that is to be added.</td>
</tr>
<tr>
<td>sentinelMaster.serviceName</td>
<td>string</td>
<td>The name of the database to be added. This is the same name used in the sentinel monitor directive.</td>
</tr>
<tr>
<td>sentinelMaster.authPass</td>
<td>string</td>
<td>(optional) The password, if any, for the monitored database. This can be different from the password of the sentinel instance itself. This is the same password as provided in the sentinel auth-pass directive.</td>
</tr>
<tr>
<td>tls</td>
<td>object</td>
<td>(optional) TLS parameters for the database</td>
</tr>
</tbody>
</table>

**Example**

```json
{
    "name": "QA Redis Sentinel DB",
    "connectionType": "SENTINEL",
    "sentinelHost": "redis-sentinel.acme.com",
    "sentinelPort": 26379,
    "sentinelPassword": "sentinel-pass",
    "sentinelMaster": {
        "serviceName": "mymaster",
        "authPass": "opensesame"
    }
}
```

**TLS parameters**

TLS parameters can be used to specify how RedisInsight should connect to the Redis database over TLS.

The following parameters can be used:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>useTls</td>
<td>boolean</td>
<td>Whether to use TLS to connect to the database or not</td>
</tr>
<tr>
<td>clientAuth</td>
<td>boolean</td>
<td>Whether TLS client authentication is required by the database</td>
</tr>
<tr>
<td>clientCertificateKeyPair</td>
<td>object</td>
<td>(optional) The details of the client certificate and private key used to connect to the Redis database. If client authentication is not required, this has to be provided</td>
</tr>
</tbody>
</table>
**verifyServerCert**

**boolean**

(optional) Whether to verify server certificate

**caCert**

**object**

(optional) The details of the CA certificate used to connect to the Redis database

---

**TLS client certificate and key**

The client certificate and key details can be provided in two forms:

1. If the certificate and key has already been used for a database before, or **has been added separately**, the ID of that certificate can be provided directly.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>number</td>
<td>The ID of the client certificate/key pair</td>
</tr>
</tbody>
</table>

2. Alternatively, to create a new client certificate/key pair for the database, a name must be provided along with the certificate and key strings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>object</td>
<td>The details of the new client certificate/key pair to be created</td>
</tr>
<tr>
<td>new.name</td>
<td>string</td>
<td>The name of the new client cert/key pair</td>
</tr>
<tr>
<td>new.cert</td>
<td>string</td>
<td>The client certificate string</td>
</tr>
<tr>
<td>new.key</td>
<td>string</td>
<td>The client private key string</td>
</tr>
</tbody>
</table>

**Example**

1. Using an existing client certificate/key pair for a new database.

   ```json
   {
       "name": "Prod Redis Enterprise DB",
       "connectionType": "STANDALONE",
       "host": "redis-ent.acme.com",
       "port": 6379,
       "tls": {
           "useTls": true,
           "clientAuth": true,
           "clientCertificateKeyPair": {
               "id": 6
           }
       }
   }
   ```

2. Creating a new client certificate/key pair while adding the database.
```json
{
  "name": "Prod Redis Enterprise DB",
  "connectionType": "STANDALONE",
  "host": "redis-ent.acme.com",
  "port": 6379,
  "tls": {
    "useTls": true,
    "clientAuth": true,
    "clientCertificateKeyPair": {
      "new": {
        "name": "Prod client certificate",
        "cert": "-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----",
        "key": "-----BEGIN PRIVATE KEY-----...-----END PRIVATE KEY-----"
      }
    }
  }
}
```

**TLS CA certificate**

The CA certificate details can be provided in two forms:

1. If the certificate has already been used for a database before, the name of that certificate can be provided directly.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the CA certificate</td>
</tr>
</tbody>
</table>

2. Alternatively, to create a new CA certificate for the database, a name must be provided along with the certificate string.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>object</td>
<td>The details of the new CA certificate to be created</td>
</tr>
<tr>
<td>new.name</td>
<td>string</td>
<td>The name of the new CA certificate</td>
</tr>
<tr>
<td>new.cert</td>
<td>string</td>
<td>The CA certificate string</td>
</tr>
</tbody>
</table>

**Example**

1. Using an existing CA certificate for a new database.
2. Creating a new CA certificate while adding the database.

```json
{
    "name": "Prod Redis Enterprise DB",
    "connectionType": "STANDALONE",
    "host": "redis-ent.acme.com",
    "port": 6379,
    "tls": {
        "useTls": true,
        "caCert": {
            "name": "my-cert"
        }
    }
}
```

Success response

**Code:** 201 Created

**Add TLS certificate and key pair**

Used to add a new TLS certificate and private key pair to use to connect to a Redis database.

**URL:** /api/tls-client-cert/

**Method:** POST

**Body Type:** JSON

**Auth required:** NO

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the new cert/key pair</td>
</tr>
<tr>
<td>cert</td>
<td>string</td>
<td>The certificate string</td>
</tr>
</tbody>
</table>
### Success response
The name and ID is returned in the response body. The ID can be used to reference this certificate/key pair when adding databases.

**Code:** 201 Created

**Body:**

```json
{
    "certificate": {
        "id": 1,
        "name": "VeriCert ACME Certificate"
    }
}
```

### Get added TLS certificate and key pairs
Used to retrieve a list of previously added TLS certificate and private key pair to use to connect to a Redis database.

**URL:** /api/tls-client-cert/

**Method:** GET

**Auth required:** NO

### Success response
A list of objects, each containing the client certificate’s name and ID, is returned in the response body. The ID can be used to reference this certificate/key pair when adding databases.

**Code:** 200 OK

**Body:**

```json
{
    "certificates": [
        {
            "id": 6,
            "name": "VeriCert ACME Certificate"
        }
    ]
}
```

**Updated:** November 28, 2021