

Redis configuration - SSO

Prerequisites

All the steps before this step in either [Windows scalability and high availability setup - SSO](#) or [Linux scalability and high availability setup - SSO](#) must have been done.

To configure SSO to use Redis backed session storage, you need to modify data in SSO configuration database (Ubilogin directory):

- Create a new ubiloginService entry in cn=Services,ou=System with following attributes:
 - ubiloginClassName *com.ubisecure.ubilogin.session.manager.redis.SessionManagerFactoryRedis*
 - ubiloginConfString url <URL of the Redis service, in form *redis://[address]:[port]>*
 - Possible to set separate entries for each seed node in the cluster, as shown in the example below.
 - The seed nodes don't need to contain all the nodes in the cluster, as SSO will discover the other nodes during runtime through Redis protocol.
 - At least one of the seed nodes needs to be online during SSO startup. Otherwise SSO fails to boot.
 - ubiloginConfString password <password for the Redis service>
- Link the created ubiloginService to the cn=ServerSession,ou=System entry using ubiloginServiceDN attribute

For example:

```
dn: cn=SessionManagerFactoryRedis,cn=Services,ou=System,@suffix@
changetype: add
objectClass: ubiloginService
cn: SessionManagerFactoryRedis
ubiloginClassName: com.ubisecure.ubilogin.session.manager.redis.SessionManagerFactoryRedis
ubiloginConfString: url redis://redisnode1.example.com:7000
ubiloginConfString: url redis://redisnode2.example.com:7000
ubiloginConfString: url redis://redisnode3.example.com:7000
ubiloginConfString: url redis://redisnode4.example.com:7000
ubiloginConfString: url redis://redisnode5.example.com:7000
ubiloginConfString: url redis://redisnode6.example.com:7000
ubiloginConfString: password SecretPassword1

dn: cn=ServerSession,ou=System,@suffix@
changetype: modify
replace: ubiloginServiceDN
ubiloginServiceDN: cn=SessionManagerFactoryRedis,cn=Services,ou=System,@suffix@
-
```

Note that @suffix@ must be expanded to the value of attribute *suffix* in win32.config.

The change can be done using for example Apache DirectoryStudio, or you can create an ldif file to change the file, and load the data using import script:

```
./ldap/openldap/import.sh ldap/[name of file containing the changes].ldif
```