

CAS 部署

1. CAS 简介

CAS — Central Authentication Service

部署

- CAS Server — CAS 部署 CAS 部署 TGT ST
- Service Management — CAS 部署 CAS 部署 CAS 部署
- CAS Client — CAS 部署 web/app 部署 CAS 部署

1.1. 部署

CAS Server

部署 cas-overlay 部署 cas-overlay 部署

<https://github.com/apereo/cas-overlay-template>

部署 cas.example.org

部署 SSL

```
keytool -genkey -alias cas -keyalg RSA -validity 999 \
-keystore /etc/cas/thekeystore -ext san=dns:cas.example.org
Enter keystore password: changeit
Re-enter new password: changeit
What is your first and last name?
[Unknown]: cas.example.org
What is the name of your organizational unit?
[Unknown]: test
What is the name of your organization?
[Unknown]: test
What is the name of your City or Locality?
[Unknown]: test
What is the name of your State or Province?
[Unknown]: test
What is the two-letter country code for this unit?
[Unknown]: CN
Is CN=cas.example.org, OU=test, O=test, L=test, ST=test, C=CN correct?
[no]: yes
```

部署/etc/hosts:

```
127.0.0.1 cas.example.org
```

#####Java#####

```
# Export the certificate into a file
sudo keytool -export -file /etc/cas/config/cas.crt -keystore /etc/cas/thekeystore
-alias cas

# Import the certificate into the global keystore
sudo keytool -import -file /etc/cas/config/cas.crt -alias cas -keystore
$JAVA_HOME/lib/security/cacerts
```

Service Management

cas-management-overlay #####

<https://github.com/apereo/cas-management-overlay>

CAS Client

cas-client

<https://github.com/apereo/java-cas-client>

Gradle#####

```
implementation 'org.jasig.cas.client:cas-client-support-springboot:3.6.0'
```

2. CAS-overlay CAS#####

CAS#####

- #####ad-hoc
- #####SpringSpring Webflow
- #####deployment#####bug#####issue

#####

- Bug#####CAS#####issuecontribute fixes#####CAS#####issue
- CAS#####CAS#####CAS#####

#####

cas-overlay #####

<https://cas.example.org:8443/cas/login>

#####/etc/cas/config/cas.properties

```
cas.authn.accept.users=sungness::sungness
```

casuser / Mellon#####

#####

#####

Application Not Authorized to Use CAS

The application you attempted to authenticate to is not authorized to use CAS. This usually indicates that the application is not registered with CAS, or its authorization policy defined in its registration record prevents it from leveraging CAS functionality, or it's malformed and unrecognized by CAS. Contact your CAS administrator to learn how you might register and integrate your application with CAS.

#####

I solved issue myself, The issue was with the serviceId not mapped correctly in the client table of the CAS database.

It took two weeks to understand the complete CAS system and resolve the issue. But Someone having the same issue can directly follow the below steps (Assuming CAS 5.1 version deployed on tomcat +linux)

Check your .json files placed under /etc/cas/services
Check your .jwks files places under /etc/cas/jwks folder
Check the clientId and ClientSecret matches in .json , .jwks and in CAS Database-Client table
Check the serviceID mapped in .json,.jwks and in your CAS Database,client table
All the best.

2.1. CAS Properties

<https://apereo.github.io/cas/6.2.x/configuration/Configuration-Properties.html>

#####CAS#####

- cas.someProperty
- cas.some-property
- cas.some_property

2.1.1.

<https://apereo.github.io/cas/6.2.x/configuration/Configuration-Properties-Common.html>

2.1.2. 属性

CAS属性

2.1.3. 自定义属性

```
cas.custom.properties.[property-name]=[property-value]
```

2.1.4. 配置

配置

配置server配置

- 属性 — `cas.standalone.configurationDirectory` 配置目录
`/etc/cas/config;/opt/cas/config;/var/cas/config`
- 属性 — `cas.standalone.configurationFile`

Spring Cloud

src/main/resources/bootstrap.properties配置

参考 <https://apereo.github.io/cas/6.2.x/configuration/Configuration-Server-Management.html>

Spring Cloud配置

- Native

```
# spring.profiles.active=native

# The configuration directory where CAS should monitor to locate settings.
# spring.cloud.config.server.native.search-locations=file:///etc/cas/config
```

- Git
- MongoDB
- Amazon S3
- JDBC
- REST

```
# cas.spring.cloud.rest.url=
# cas.spring.cloud.rest.basic-auth-username=
# cas.spring.cloud.rest.basic-auth-password=
# cas.spring.cloud.rest.method=
# cas.spring.cloud.rest.headers=Header1:Value1;Header2:Value2
```

2.1.5. 数据库

数据库 <https://apereo.github.io/cas/6.2.x/configuration/Configuration-Properties-Security.html>

2.1.6. 数据库

- 数据库
- 数据库CPU数据库
- session数据库ticketsSSO session数据库
- JDBC数据库
- LDAP 数据库
- MongoDB数据库

3. Service Management

<https://apereo.github.io/cas/6.3.x/services/Service-Management.html>

CAS数据库CAS数据库CAS数据库CAS数据库service registry

- Authorized services - 数据库CAS SSO session
- Forced authentication - 数据库
- Attribute release - 数据库
- Proxy control -
- Theme control - 数据库CAS

service-registry数据库

- cas.service-registry.*

数据库

<https://apereo.github.io/cas/6.3.x/configuration/Configuration-Properties.html#service-registry>

3.1. 数据库

- registeredServices — 数据库GETDELETE
- exportRegisteredServices — ZIP数据库
- importRegisteredServices — 数据库

3.2. 数据库Web

webapp数据库web数据库CAS数据库GUI数据库web数据库CAS server数据库

webapp

<https://apereo.github.io/cas/6.3.x/services/Installing-ServicesMgmt-Webapp.html>

Overlay Installation

<https://github.com/apereo/cas-management-overlay>

3.3. Registered Services

- `id` —
- `name` — 255
- `description` — 255
- `informationUrl` — URL
- `privacyUrl` — URL
- `redirectUrl` — URL
- `logo` — CAS web images
- `serviceId` — logical service URLs URL
- `theme` — ticket CAS UI values Spring Expression Language
- `proxyPolicy` —
- `evaluationOrder` — URL
- `authenticationPolicy` — engine
- `attributeReleasePolicy` —
- `logoutType` — logout
 - `LogoutType.BACK_CHANNEL`
 - `LogoutType.FRONT_CHANNEL`
 - `LogoutType.NONE`
- `responseType` — CAS
- `usernameAttributeProvider` — “username”
- `accessStrategy` —
- `publicKey` —
- `logoutUrl` — logout URL
- `properties` — key/value
- `multifactorPolicy` —
- `contacts` —
- `matchingStrategy` —

Service Type

[illegible]

3.3.1. □□□□□□

3.3.2. □□□□□□

3.3.3. Required Authentication

3.3.4. Tags & Properties

3.3.5. Contacts & Owners

3.3.6. □□□□

3.3.7. □□□□

3.3.8 □□

CAS 0000000000000000

- Memory — XML CAS
- JSON — JSON HA
- YAML — YAML HA
- GIT — GIT HA
- MongoDB — MongoDB HA
- Redis — Redis HA
- LDAP — LDAP HA
- JPA — JPA HA
- Couchbase — Couchbase HA
- CouchDB — CouchDB HA
- ...
- REST — REST API HA
- Custom — CAS APIs HA

Memory□□□□□□□□

Spring Bean webapp UI

```
@Configuration("myConfiguration")
public class MyConfiguration {

    @Bean
    public List inMemoryRegisteredServices() {
        final List services = new ArrayList<>();
        final RegexRegisteredService service = new RegexRegisteredService();
        ...
        services.add(service);
        return services;
    }
}
```

JSON

classpath:src/main/resources/services

JSON

```
implementation "org.apereo.cas:cas-server-support-json-service-
registry:${casServerVersion}
```

JSON

```
# cas.service-registry.json.location=classpath:/services
```

JSON fileName = serviceName + "-" + serviceNumericId + ".json"

YAML

```
implementation "org.apereo.cas:cas-server-support-yaml-service-
registry:${casServerVersion}
```

YAML

```
# cas.service-registry.yaml.location=classpath:/services
```

YAML fileName = serviceName + "-" + serviceNumericId + ".yaml"

