Directories & Security 6.5

Revised: June 22nd, 2020

Directories used by Couchbase

- For executables: /opt/couchbase/bin
- For man pages : /opt/couchbase/share/man
  - man1,man4,man7
- For tmp space for queries: /opt/couchbase/var/lib/couchbase/tmp
- For logs : /opt/couchbase/var/lib/couchbase/logs
- For stats : /opt/couchbase/var/lib/stats/

Parental Directories for Services

- Services [data, index, query, eventing,analytics,] :
  /opt/couchbase/var/lib/couchbase/data (default)
  - Recommend the following changes……
    - **Data:** /opt/couchbase/var/lib/couchbase/data
    - **Index:** /opt/couchbase/var/lib/couchbase/indexes
    - **Query:** /opt/couchbase/var/lib/couchbase/query
    - **Eventing:** /opt/couchbase/var/lib/couchbase/eventing
    - **Analytics:** /opt/couchbase/var/lib/couchbase/analytics

Directories and files

- Sample databases : /opt/couchbase/samples
  - beer-sample, travel-sample, gamesim-sample
- Auditing: /opt/couchbase/var/lib/couchbase/logs/audit.log
- cbcollect_info default output directory:

Why Security?

The Net is Dark and Full of Terrors

Recent Security Breaches

Iran
Russia
China

WannaCry Ransomware

Wikileaks CIA Vault 7

Cloudbleed

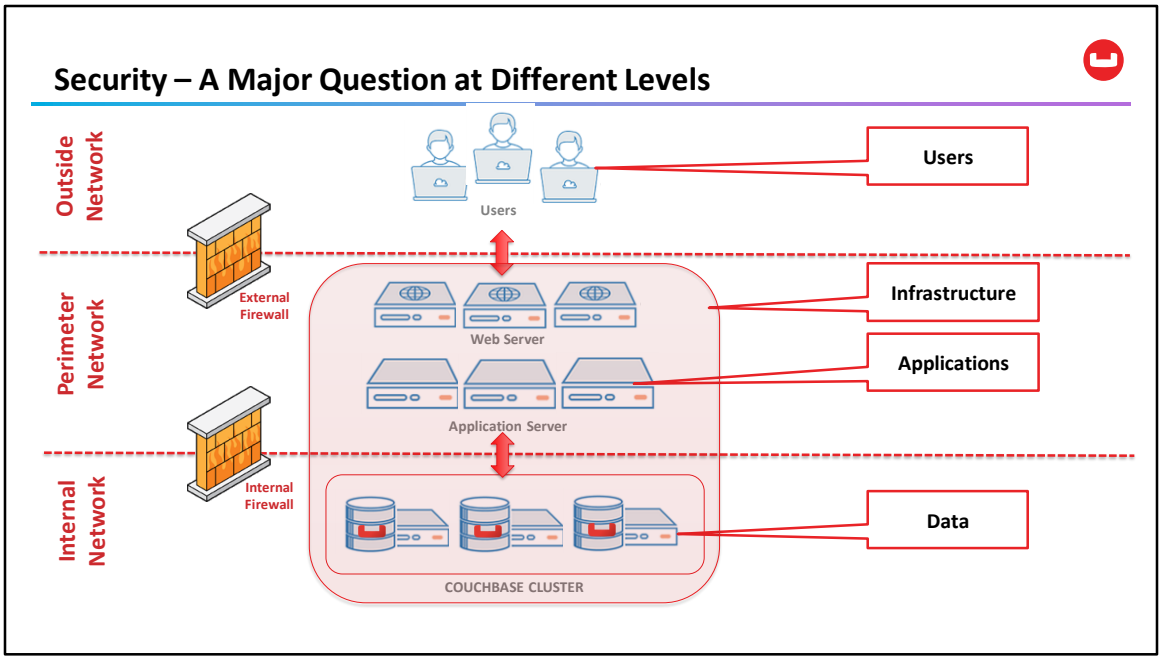
Big Company ?
Who's Next ??

Every Year……

Objective

- **Quick review of security capabilities**
- **Authentication**
  - PAM authentication in Couchbase
- **Authorization**
  - Role Based Access Control for Applications
- **Cryptography**
  - Secret Management for Couchbase
- **Security Roadmap**

Security Capabilities – A quick refresher



Where Security is Enforced

Some applications need an additional layer of security to meet business or regulatory compliance requirements. In nearly all commercial deployments of Couchbase, Couchbase is deployed on a trusted network, and unauthorized access is restricted by firewall routing rules.

From the network perspective, here are a few layers you might consider for enforcing security:

Outside network, where web browsers and mobile applications are located.
Perimeter network between the internal and external firewall, which typically consists of web servers and load balancing machines. This network provides physical separation between back-end and external interfaces, such as the web and mobile applications.
Internal network within the internal firewall, where Couchbase Server is typically deployed.

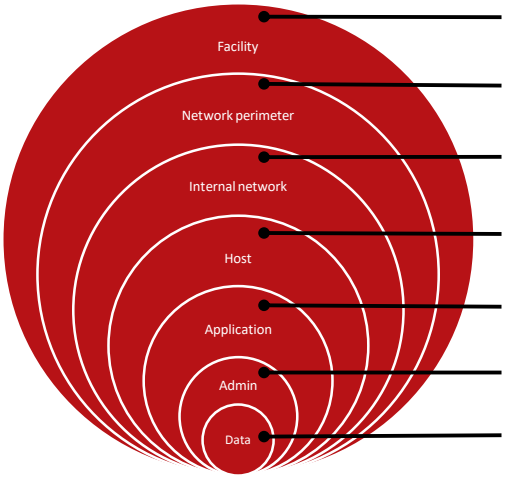
Requests from the external network come through an external firewall and are

directed to the load balancing unit, where security administrators can introduce packet filtering and blocking of malicious IP addresses. After that, the requests proceed to a web server.

On the second firewall level, between the perimeter and internal network, the IT or database administrators can allow only Couchbase ingress and egress ports to be accessible through the internal firewall.

While the external firewall allows only certain ports to be open, the internal firewall allows only certain Couchbase ports to be open.

Defense in Depth:
Multi-dimensional approach to customer environment



- Physical controls, video surveillance, access control
- Edge routers, firewalls, intrusion detection, vulnerability scanning
- Intrusion detection, vulnerability scanning
- Access control and monitoring, anti-malware, patch and configuration management
- Secure engineering (SDL), **Access Control**, security monitoring, anti-malware
- **Account Management**, training and awareness, screening
- Authorization, Data Encryption, Data Masking, **Secret Management**

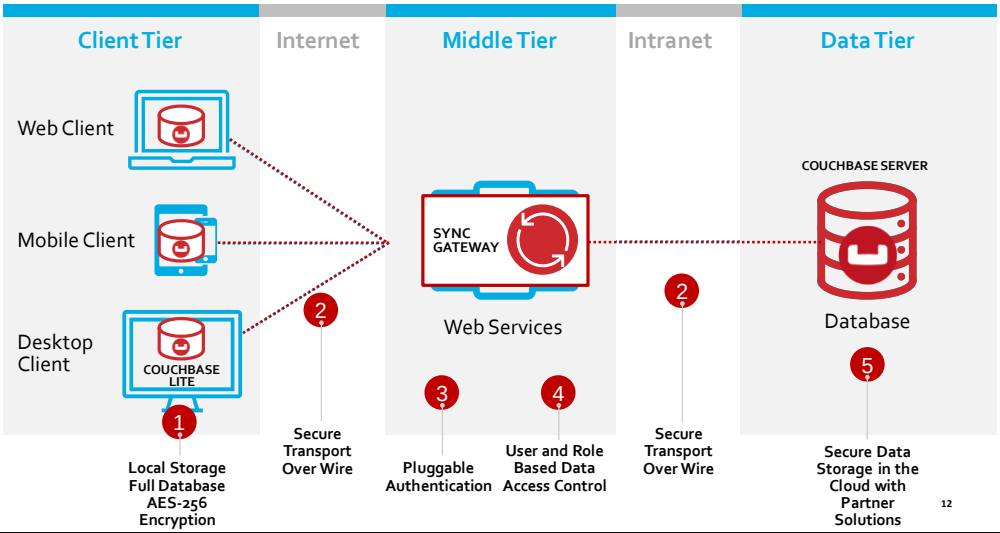
Security Pillars in Couchbase

Authentication	Authorization	Crypto	Auditing	Operations
App/Data: SASL AuthN Admin: Local or LDAP **PAM Authentication (4.6)**	Local Admin User Local Read-Only User RBAC for Admins **RBAC for Applications (5.0)** **LDAP groups(6.5)**	TLS for admin access,client-server access, Secure XDCR X.509 certificates for TLS Data-at-rest Encryption* Field-level Encryption** **Secret Management (4.6)** **Node to Node encryption(6.5)**	Admin auditing	Security management via UI/CLI/REST

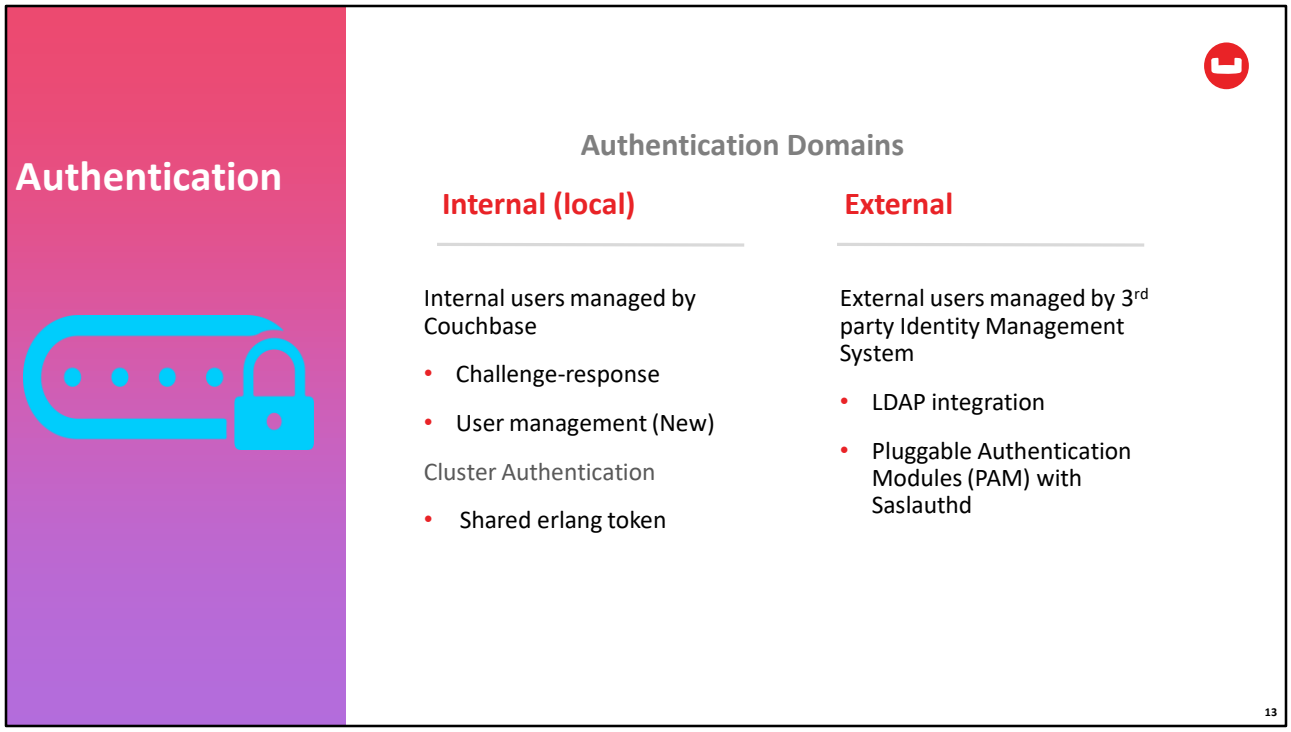
SASL = Simple Authentication and Security Layer
TLS = Transport Layer Security

*** Via third-party partners** 3rd partners (Vormetric, Protegrity, SafeNet) **** via Couchbase pakage**

Couchbase addresses Security concerns for the full stack



12



Authentication Domains

Couchbase Server assigns users to different authentication domains:

Local: Contains users defined locally. This includes:

The Full Administrator for Couchbase Server.

Internal Components within Couchbase Server that support core functionality (for example, indexing, searching, and replicating), and run with full administrative privileges.

Generated Users, which are created by Couchbase Server as part of the upgrade process from pre-5.0 to 5.0 and post-5.0 versions; each in correspondence with a legacy bucket. Each Generated User is assigned a username that is identical to the bucket-name; and either a password that is identical to the bucket's pre-5.0 password, or no password, if the bucket did not feature a password. Generated Users are created to ensure that legacy applications can continue to access legacy buckets after upgrade to 5.0 or post-5.0, with the same username-password combination

being used for authentication.

Locally Defined Users, which are explicitly created by a Couchbase Server administrator; and each feature a username and password unique within the Local domain.

External: Contains users defined externally; either by means of LDAP or PAM. Passwords are defined and stored remotely. Note that External usernames do not clash with Local usernames.

When a user attempts to authenticate, Couchbase Server always looks up their credentials in the same order: which is Local first, and External second.

Authorization

Authorization for Admins

- **Role based access control for Administrators**

Authorization for Apps

- **RBAC for applications**

Including LDAP for Groups mapping(ver 6.5)

14

Role-Based Access Control

Couchbase provides Role-Based Access Control (RBAC), in which access privileges are assigned to fixed roles; which are in turn assigned to administrators and applications.

Couchbase Server Enterprise Edition provides RBAC with multiple roles for finer access control.
Community Edition provides multiple users that can be assigned to limited set of roles.

There are three fixed roles in the community edition of Couchbase providing coarser access control:
Bucket Full Access (bucket_full_access[*]), Admin (admin), and Read Only Admin (ro_admin).

Role-Based Access Control (RBAC) for Administrators

Role-Based Access Control (RBAC) allows you to specify what each admin can access in couchbase through role membership

Regulatory Compliance

A strong demand for applications to meet standards recommended by regulatory authorities

Segregation of Admin Duties

Every admin does not have all the privileges. Depending on the job duties, admins can hold only those privileges that are required.

Security Privilege Separation

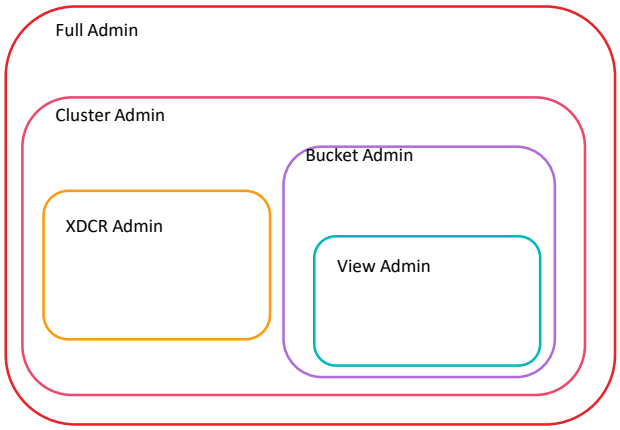
Only the full-admin has the privilege to manage security, and his/her actions can be audited just like other administrators.

RBAC for Administrators – How it works

- Administrative users can be mapped to out-of-the-box roles
- Roles pre-defined with permissions for specific resources
  - Full Admin
  - Cluster Admin
  - Bucket Admin
  - View Admin
  - XDCR Admin
- Can work with internal and external users

Roles can be single or combined

and mapped to an external LDAP group (ver 6.5)



Encryption

On-the-wire Encryption

- TLS between client and server
- TLS between datacenters using secure XDCR
- X.509 CA Certificates for trusted encryption between client and server

On-Disk Encryption

- Volume and application level encryption through our trusted 3rd partners (Vormetric, Protegrity, SafeNet)
- FIPS 140-2 compliant

Encryption in Applications

- Encrypt data inside your application, even before it is sent to the server. This allows encryption to be used selectively, for only the most sensitive information you intend to send.

Secret Management

- Secret-Management provides a way of managing server-secrets. This increases the security of your data, and potentially makes it easier to meet compliance-requirements.

https://developer.couchbase.com/documentation/server/5.5/security/secret-mgmt.html

Role Based Access Control –

Role-Based Access Control (RBAC) for Applications

- Meet regulatory compliance requirements for data users and applications
- Simplified access control management for data and admin users across the cluster

Regulatory Compliance

A strong demand for applications to meet standards recommended by regulatory authorities

Segregation of User Duties

Depending on the job duties, users can hold only those privileges that are required

Locking Down Services

Depending on what the service is needed for, only those roles can be assigned

RBAC Security Model

Privilege

A set of actions on a given resource
Eg. Read documents on "foo" bucket

Role(s)

A fixed grouping of privileges that defines the access given

User

User is a human user or service

Action: an operation *eg. read, write, read metadata*

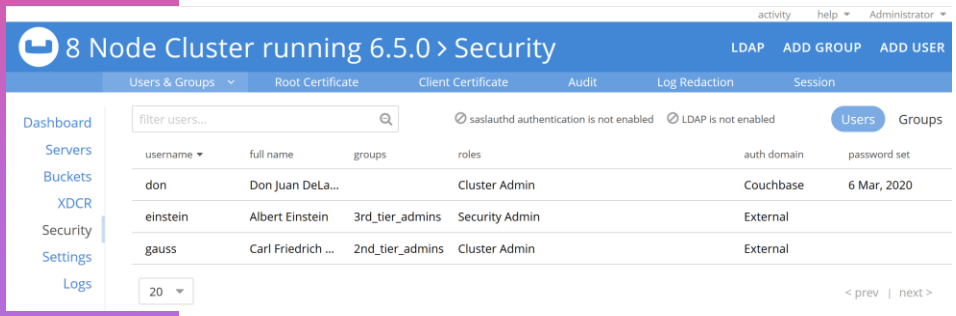
Resource: some system object that an action can be performed on. *eg. bucket, index, etc.*

- NIST Model
- Scalable users accounts
- Fixed out-of-the-box data roles in 5.0
- 1:N User-to-role mapping
- Roles can be applied for specific buckets / across all buckets [*]

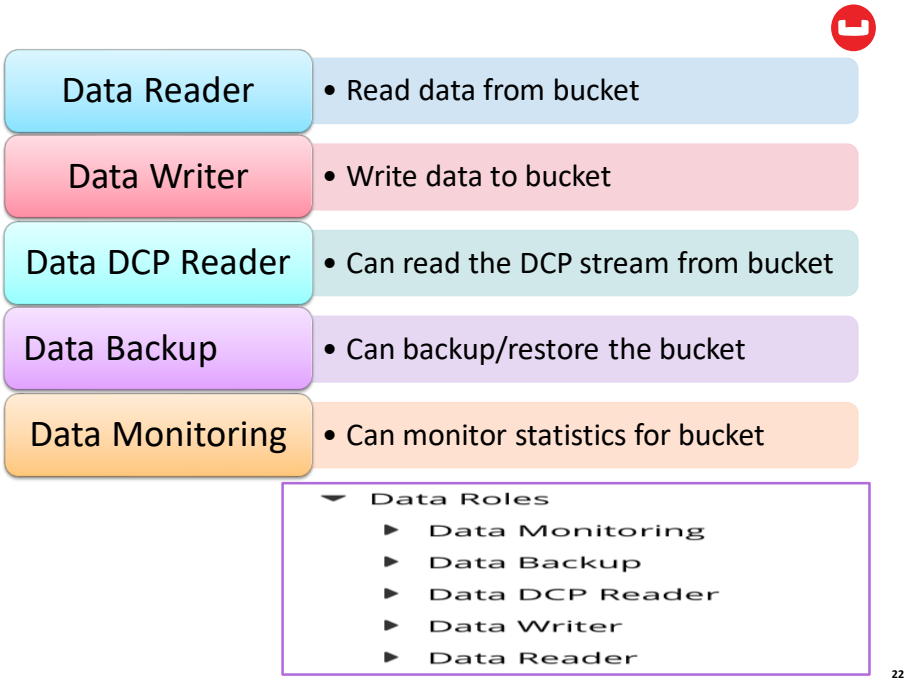
User Management

Flexible User Management

- Internal and External authorization support
- Unique identities for data users and services
- REST and CLI configurable
- Seamless upgrades without application changes
- Scalable



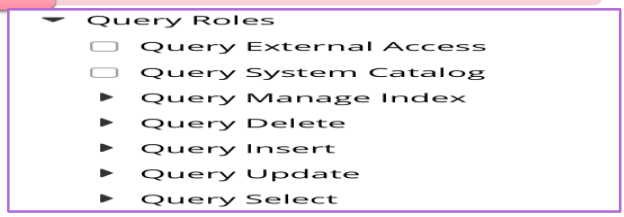
Roles for Data Service



We have added 5 new data roles

Roles for Query Service

Query Select	• Can execute SELECT N1QL statement for bucket
Query Update	• Can execute UPDATE N1QL statement for bucket
Query Insert	• Can execute INSERT N1QL statement for bucket
Query Delete	• Can execute DELETE N1QL statement for bucket
Query Manage Index	• Can execute index management statements for bucket
Query System Catalog	• Can query system tables for bucket
Query External Access	• Can execute N1QL CURL statement

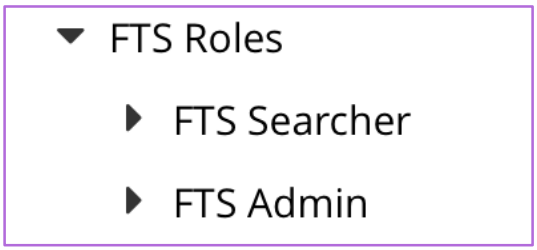


23

We have added 7 query roles

Roles for Full Text Search Service

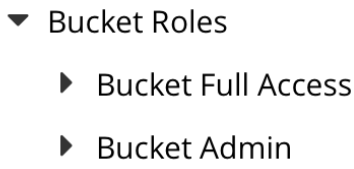
FTS Admin	• Can administer FTS service
FTS Searcher	• Can execute search queries for a bucket



Bucket Roles

So, can I get a role that gives me the application behavior similar to pre-5.0?

Bucket Full Access	• Full Read/Write access over the bucket
Bucket Admin	• Full Read/Write access over the bucket, and ability to change bucket settings



Password Policy and Rotation

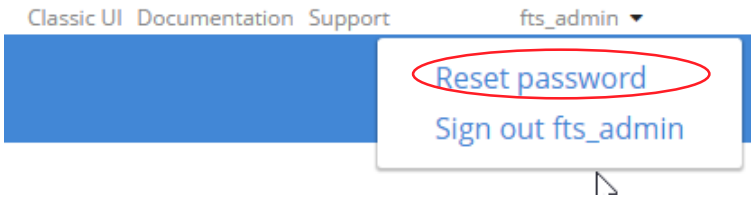
Default Policy

```
{
 "enforceDigits": false,
 "enforceLowercase": false,
 "enforceSpecialChars": false,
 "enforceUppercase": false,
 "minLength": 6
}
```

Policy and Rotation

- Simple password policy rules enforced when initially set or rotated
- Policy can be set using REST or CLI
- Password can be reset using UI, REST or CLI

couchbase-cli setting-password-policy [--cluster <url>] [--username <user>] [--password <password>] [--get] [--set] [--min-length <num>] [--uppercase] [--lowercase] [--digits] [--special-chars]



26

SYNTAX

couchbase-cli setting-password-policy [--cluster <url>] [--username <user>]
[--password <password>] [--get] [--set] [--min-length <num>] [--uppercase]
[--lowercase] [--digits] [--special-chars]

```
$ couchbase-cli setting-password-policy -c 192.168.1.5 -u Administrator \
 -p password –get
```

```
$ couchbase-cli setting-password-policy -c 192.168.1.5 -u Administrator \
 -p password --set --min-length 10 --uppercase --lowercase --digits
```

Role Assignment – Using REST and CLI

Using REST

```
curl -X PUT http://localhost:8091/settings/rbac/users/local/don-data-user
 -u Administrator:password -d "roles=data_reader[travel-sample]" -d "password=donpassword”
```

Using CLI

```
couchbase-cli user-manage --set --rbac-username don-n1ql-user --rbac-password donpassword --auth-domain local --roles "data_reader[*], query_select[*]" -c http://localhost:8091 -u Administrator -p password
```

GRANT /REVOKE statements in N1QL for RBAC

GRANT ROLE

GRANT ROLE data_reader(`*`) to don

REVOKE ROLE

REVOKE ROLE data_reader(`*`) from don

New system tables for RBAC

system:applicable_roles (provides user-role mappings)

```
SELECT * FROM system:applicable_roles
WHERE bucket_name="travel-sample"
```

system:user_info (provides full user information)

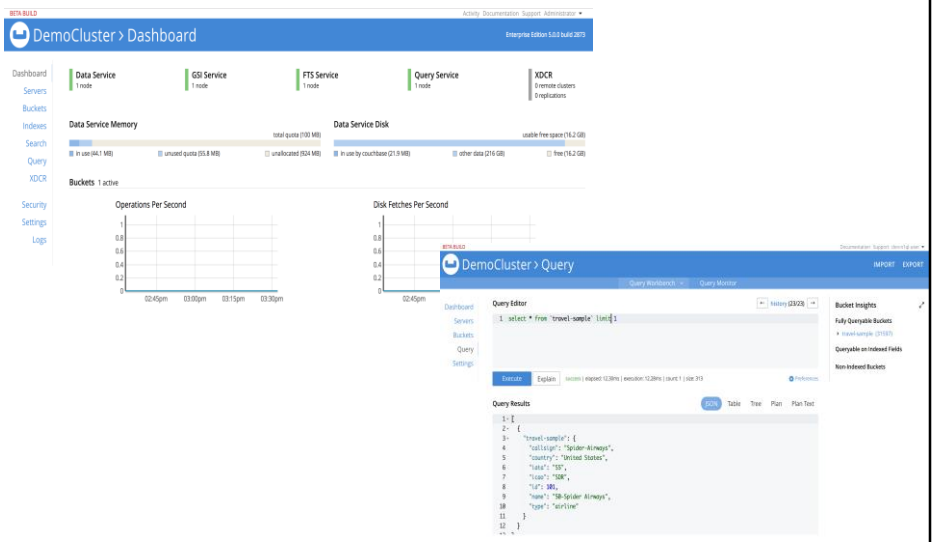
```
SELECT * FROM system:user_info
```


System:applicable_roles, system:user_info

Web Console For Administrators and Developers

Who gets to log into web console ?

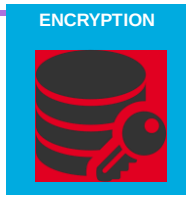
1. Administrators (Any administrator role)
2. Developers (Users who have one or more query role)



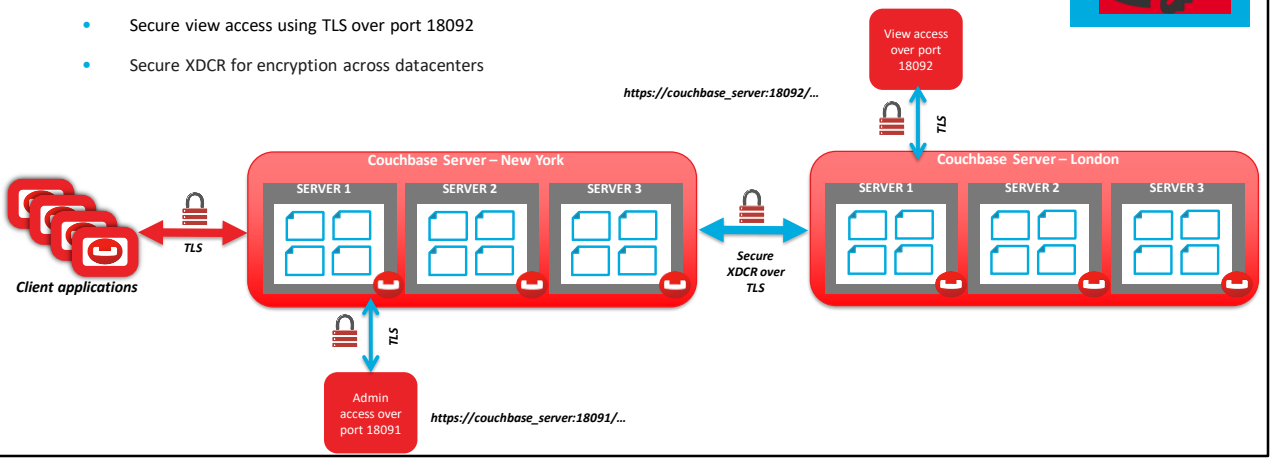
couchbase-cli user-manage --set --rbac-username don-n1ql-user2 --rbac-password donpassword --auth-domain local --roles "data_reader[*], query_select[*]" -c http://localhost:8091 -u Administrator -p password

Encryption

Couchbase encryption overview (In Motion)

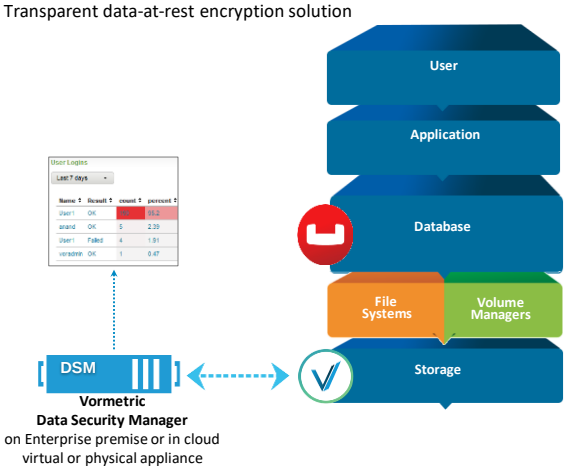


- **Data-in-motion encryption**
  - Client-server communication can be encrypted using TLS
  - Secure admin access using TLS over port 18091
  - Secure view access using TLS over port 18092
  - Secure XDCR for encryption across datacenters



Couchbase encryption overview

- Transparent data-at-rest encryption solution



ENCRYPTION

Secure Personally Identifiable Information
- User profile information
- Login Credentials
- IP Addresses

- Centrally manage keys and policy
- Virtual and physical appliance
- High-availability with cluster
- Multi-tenant and strong separation of duties
- Proven 10,000+ device and key management scale
- Web, CLI, API Interfaces
- FIPS 140-2 certified

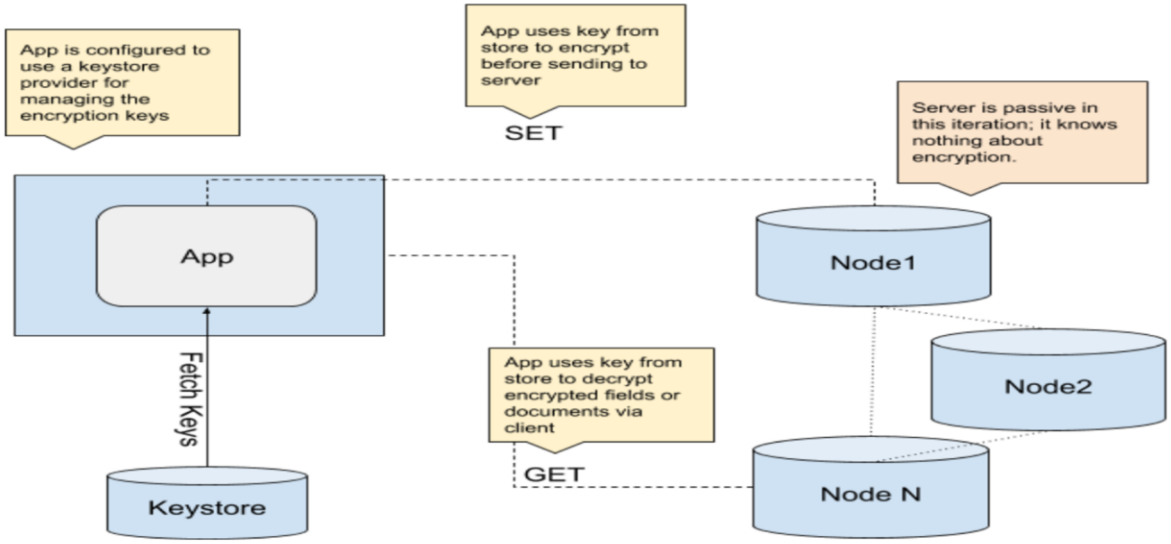
Field Level Encryption

Field Level Encryption (FLE)

- The encryption and decryption must happen entirely on the client; the server is completely passive.
- Provides encryption of JSON fields "in transit" and "at rest"
- Packaged separately from the SDK - no SDK API change
- Requires some configuration/setup to be used by application
- Cross SDK compatible
- Supports various platform idiomatic key stores
- Couchbase Enterprise licensed

Currently only K/V supported
For other services a solution is to do a query and fetch keys and then encrypt/decrypt fields directly using public API

Field Level Encryption: How it works



Field Level Encryption: JSON

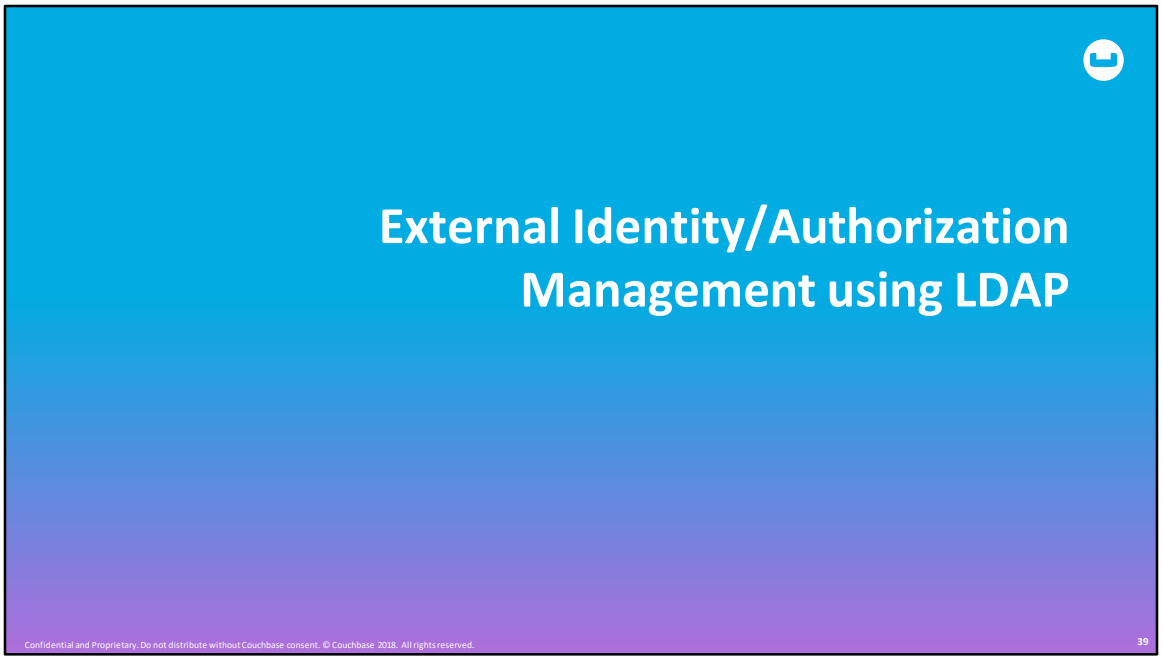
```
{
"message":"The old grey goose jumped over the wrickety gate.",
“recipient”: “jeffry.morris@couchbase.com”
}
```

```
{
 "__crypt_message": {
  "alg": "RSA-2048-OAEP-SHA1",
  "kid": "MyPublicKeyName",
  "ciphertext":
"iX2MXbUlief8Xxk4DYysivEsUXeoiFBLkm4/EC7E9vRnGikDOiuaWllLTJU/oN
KeVNlWPzfN6r/uLEpttp+BLC0DswdxLkA3ONeO85TDdHaHmrJ3dJQ7qgDFe
35K6MbTEPXE98f1wL2vOL70xJxW+3KsgdcYYYqg8VNw2U9eKVC2lv4DS19l
/r+6l+O8EGvBaa0FidezgF7CzgdXpGmG20cA0D8yCmmGoW8oq7KWoq0P
NaKsb9JOYfOYi13bxpPOIbyI003qLb5b7y1qVms8KDZ0+nk7Xnn5OYFmBHQ
DyJ39nuibEMKNMlA2ZNICvfFqE1dU3iqqZYyS7OTukFBO2g==“
 },
“recipient”: “jeffry.morris@couchbase.com”
}
```

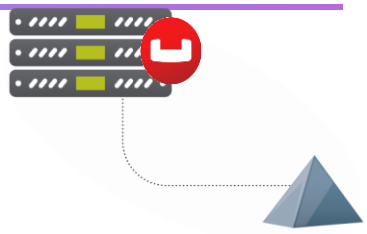
- Why not XATTRs? You can only access 1 XATTR at time, other services are (view, fts) do not have XATTR support, and cannot enumerate them (one XATTR at a time), because it is data and not meta-data; its just encrypted data.
- What is an Alg, what is a Kid?
  - Loosely follows JOSE: Json Object Signing and Encryption field names.
- Why do we add so much metadata to the document?
- Only top level fields can be encrypted
  - Customers should be directed to place all data to be encrypted into a single, top level field
- (from Matt) Users will need to audit crypto, and the prescribed format allows for simple auditing through N1QL. Downside: lots of metadata, which can be simplified in the future.

Field Level Encryption: Algorithms

- Currently supported algorithms
  - RSA-2048
    - Keysize: 2048
    - Padding: OAEP-SHA1
  - AES-256
    - Key size: 256
    - Padding: PKC7S
    - Cipher mode: CBC
    - Block size: 128
    - IV size: 16S
- API is extensible (for some SDKs)
- Algorithms are FIPS 140-2 compliant
- ICryptoProvider interface implementations


External Identity/Authorization Management using LDAP
Confidential and Proprietary. Do not distribute without Couchbase consent. © Couchbase 2018. All rights reserved.
39

External identity management using LDAP



- **Centralized identity management**
  - Define multiple read-only admins and full-admins
  - Centralized security policy management for admin accounts for stronger passwords, password rotation, and auto lockouts
- **Individual accountability. Simplified compliance.**
  - Define UIDs in LDAP, and map UIDs to read-only / full admin role in Couchbase
  - Comprehensive audit trails with LDAP UIDs in audit records

Why authentication with LDAP and why is it important?

Well, if your enterprise is already using LDAP, and you have defined users, trees, and hierarchies in LDAP, you could leverage these same identities in Couchbase environment too.

And also, you could map multiple admins that you might have defined in your LDAP server to Couchbase Read/Full Admin user, which allows you again to use the same existing policies that you have defined, such as stronger password enforcement, password rotation and also password attempt lock outs still apply to Couchbase environment too.

In addition to this, you would now be able to get individual accountability as you will not be using single Administrator anymore but rather be using multiple users that are mapped to Couchbase Admin, and be able to get audit trail on Individual user ID's.

6.5.0 Feature Overview

- What is this new capability?

  Native LDAP integration. LDAP groups support.

- When and why would a customer use it?

  It makes sense to use it if the customer uses centralized LDAP server for user management.

- What are the key functional / performance benefits it brings

  LDAP groups.

  Supports slapd and Active Directory

  Works in all platforms: Linux, MaxOSX, Windows

Feature Overview

- What use cases does this *not* fit well for?

  Might not be very convenient to use it if a customer wants to map too many (like thousands) ldap groups to Couchbase roles. We assumed that the customer might have many groups in LDAP, but will want to use only some of them in Couchbase.

  The concern is connected to the fact that in order to use the LDAP group a separate CB group must be created and mapped to a Couchbase group.

Feature Operation

LDAP configuration consists of four parts:

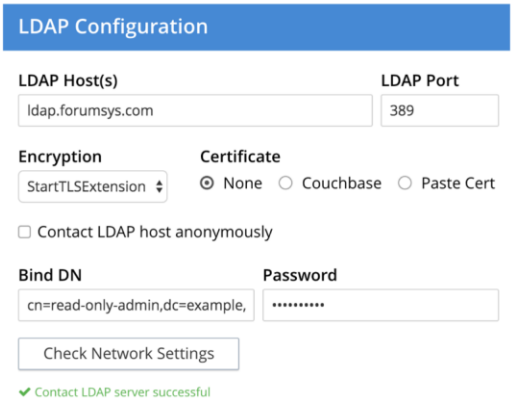
1) LDAP server connection settings (mandatory)
   - LDAP Server addresses and port
   - Encryption (+ optionally certificate for server name verification)
   - Bind DN and password
2) Username to LDAP DN mapping configuration (mandatory)
   - Construct DN using template
   - Or LDAP DN search settings
3) LDAP groups search configuration (optional)
   - Groups search settings
4) Advanced settings (optional - defaults should be ok most of the time)
   - Timeouts
   - Cache settings
   - Nested groups support settings

Feature Operation - configuration example

1) LDAP server connection settings

Notes:

- "Certificate" here means CA certificate for server name verification;
- If Certificate is "None", no verification will be done;
- "Couchbase" certificate means "use the same CA as this cluster is using";
- Connection and credentials can be tested by pressing "Check Network Settings"

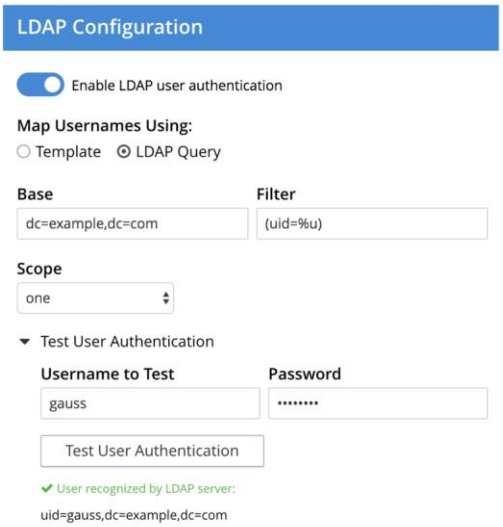


Feature Operation - configuration example

2) Mapping of usernames to LDAP DN

Notes:

- %u is replaced with username;
- Authentication can be tested by pressing “Test User Authentication”;
- DN for provided user is shown if user is found. In this case “gauss” is mapped to “uid=gauss,dc=example,dc=com”.



Feature Operation - configuration example

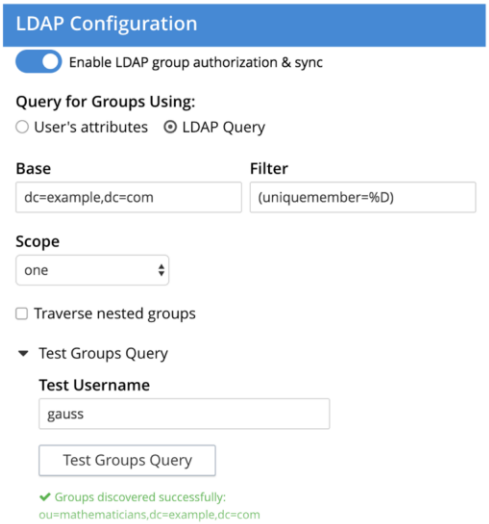
3) LDAP groups search configuration

%D is replaced with user's DN (%u is also available).

Depending on LDAP hierarchy there are two different search types possible:

- When user contains groups in attributes (usually 'memberOf' attribute);
- When group contains users in attributes (usually 'member' attribute);

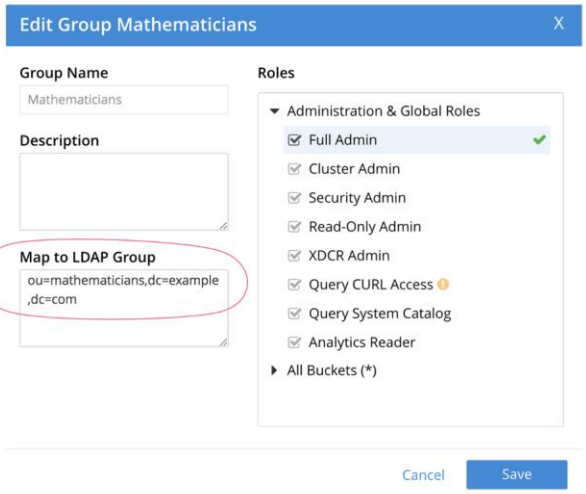
Groups search can be tested by pressing "Test Groups Query". In this case there is only one group found for 'gauss': 'ou=mathematicians,dc=example,dc=com'



Feature Operation - configuration example

When LDAP is configured LDAP groups can be associated with CB groups.

No need to create external CB users for LDAP users in this case. LDAP users will get CB roles via “LDAP groups ↔ CB group” association.

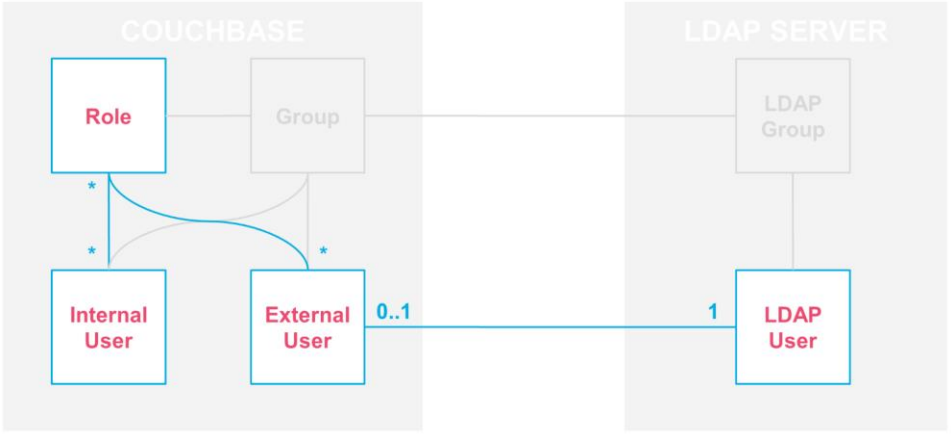


Feature Operation - configuration example CLI example:

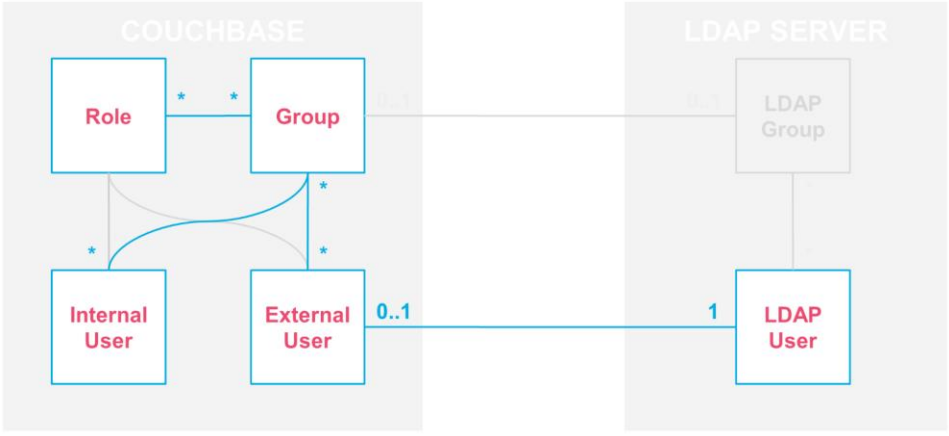
```
[ec2-user@Couchbase01 ~]$ setting-ldap -c 127.0.0.1:8091 -u Administrator -p --hosts ldap.forumsys.com
--port 389 --encryption startTLS --server-cert-validation 0
--bind-dn cn=read-only-admin,dc=example,dc=com --bind-password password --user-dn-query
"dc=example,dc=com??one?(uid=%u)" --authentication-enabled 1 --authorization-enabled 1
--group-query "dc=example,dc=com??one?(uniquemember=%D)"

$ couchbase-cli setting-ldap -c 127.0.0.1:8091 -u Administrator -p couchbase
--get | python3 -mjson.tool
```

```
{
    "authenticationEnabled": true,
    "authorizationEnabled": true,
    "hosts": [
        "ldap.forumsys.com"
    ],
    "port": 389,
    "encryption": "StartTLSExtension",
    "userDNMapping": {
        "query": "dc=example,dc=com??one?(uid=%u)"
    },
    "bindDN": "cn=read-only-admin,dc=example,dc=com",
    "bindPass": "**********",
    "groupsQuery": "dc=example,dc=com??one?(uniquemember=%D)",
    "serverCertValidation": false,
    "nestedGroupsEnabled": false,
    "requestTimeout": 5000,
    "maxParallelConnections": 100,
    "maxCacheSize": 10000,
    "cacheValueLifetime": 300000,
    "nestedGroupsMaxDepth": 10,
    "failOnMaxDepth": false
}
```

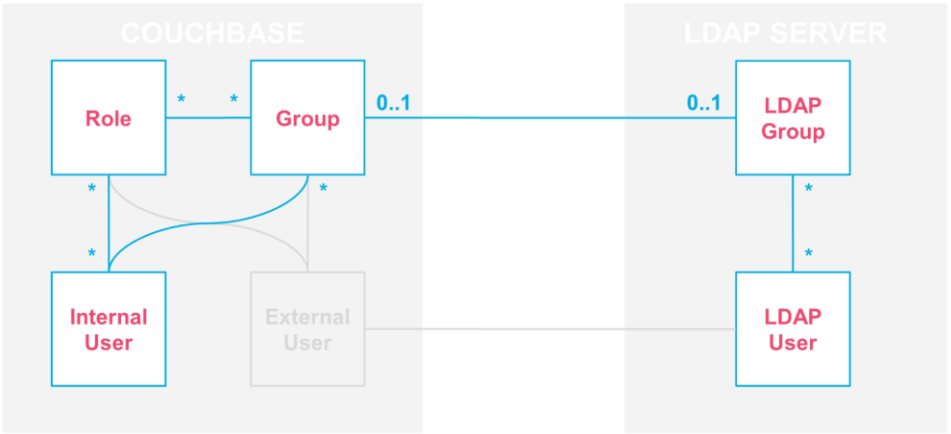
USE CASE 1: LDAP Authentication and no groups

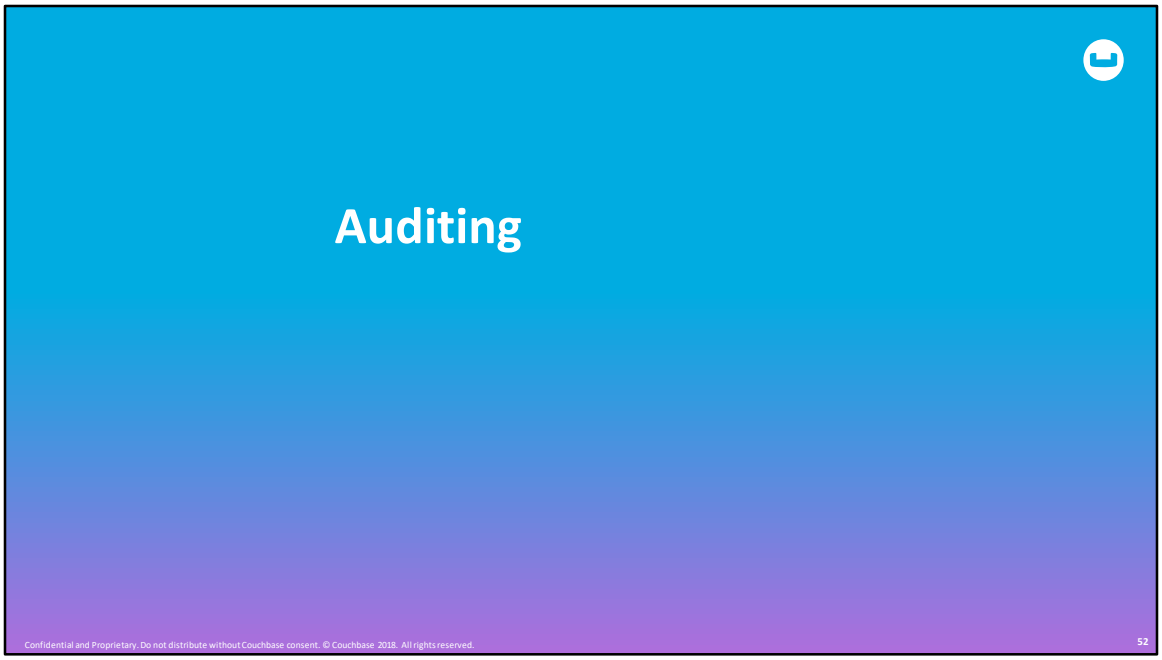


USE CASE 2: LDAP Authentication and CB Groups



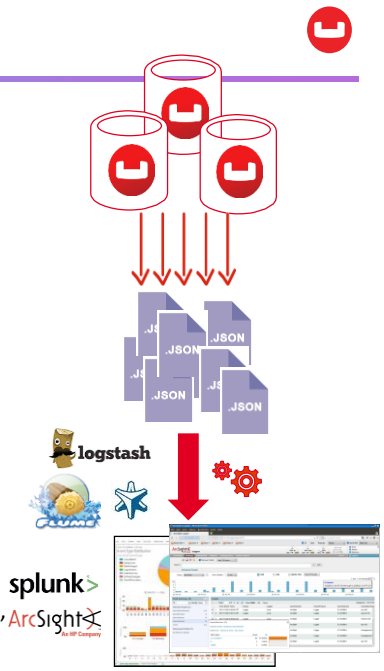
USE CASE 3: LDAP Authentication and LDAP Groups




Auditing
Confidential and Proprietary. Do not distribute without Couchbase consent. © Couchbase 2018. All rights reserved.
52

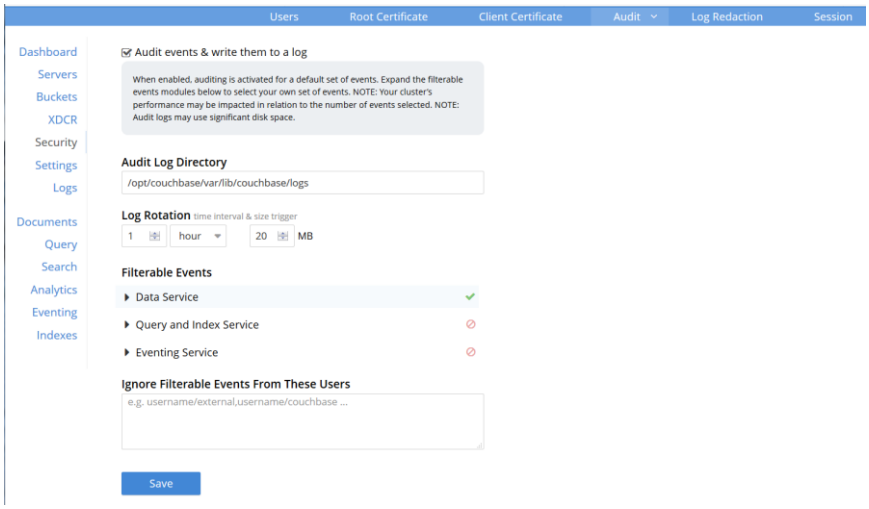
Admin Auditing in Couchbase

- **Rich audit events**
  - Over 25+ different, detailed admin audit events
  - Auditing for tools including backup
- **Configurable auditing**
  - Configurable file target
  - Support for time based log rotation and audit filtering
- **Easy integration**
  - JSON format allows for easy integration with downstream systems using flume, logstash, and syslogd



Now that you have integrated LDAP authentication, what about Auditing? Most of the enterprises now require auditing to meet regulatory & compliance purposes. Some of our customers use this to find out unsuccessful logins and basically blacklist those IP's.
Couchbase provides Rich audit events, over 25+ different, detailed audit events.
This could be configured as well, lets you want to send this log to a remote machine where only authorized personal have access to this audit data, may be a government agency or regulatory board.
We also support rotation, similar to log rotation you can configure to rotate everyday, week or a month, and also you could filter these events based on your use case, lets say you are only interested in logins & login failures, you could just filter these out and you might not be interested in who created the bucket or who deleted the bucket.
Audit events are written in JSON format, again the choice of JSON is for easy integration with downstream systems, like you might already be using splunk or logstash for rich dashboard kind of reporting, and they natively support JSON.

Auditing



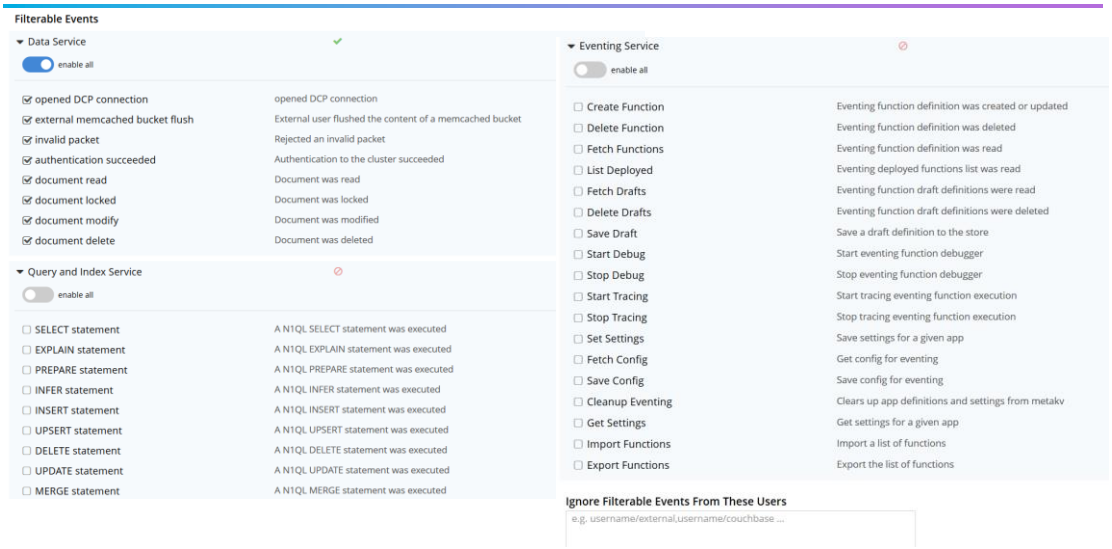
This makes the default pathname within the Audit Log Directory text-field editable. For Linux, the pathname is /opt/couchbase/var/lib/couchbase/logs; for Windows, C:\Program Files\Couchbase\Server\var\lib\couchbase\logs; for MacOS, /Users/couchbase/Library/Application Support/Couchbase/var/lib/couchbase/logs.

If you wish to modify the pathname, enter the appropriate content. Records will be saved to the directory you specify. Note the advisory message now visible beneath the checkbox: as this indicates, electing to audit a wide range of events may significantly impact performance and consume disk-space.

The Log Rotation time interval & size trigger determines at what times stored log files — referred to as targets — are rotated: this means that the current default file, to which records are being written, named audit.log, is saved under a new name, which features an appended timestamp. For example: usermachinename.local-2017-03-16T15-42-18-audit.log.

The number of time-units is specified by changing the number 1, which appears in the interactive field by default. The time-unit type is specified by means of the pull-down menu, at the right-hand side of the field:

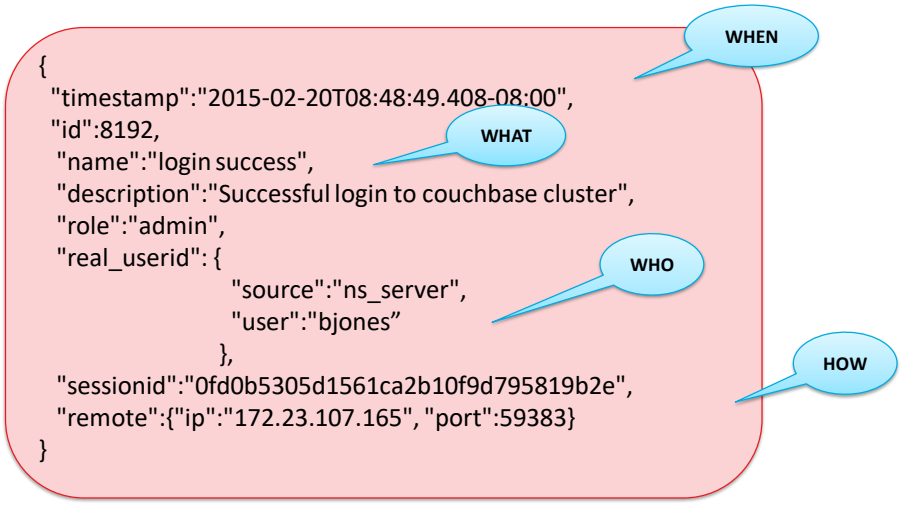
Auditing - Filterable Events



Events can be filtered for the Data Service, the Query and Index Service, and the Eventing Service. Filtering means selective logging.
Every checkbox appears selected, indicating that each corresponding event will be logged. To de-select individual events, simply uncheck the appropriate checkboxes.

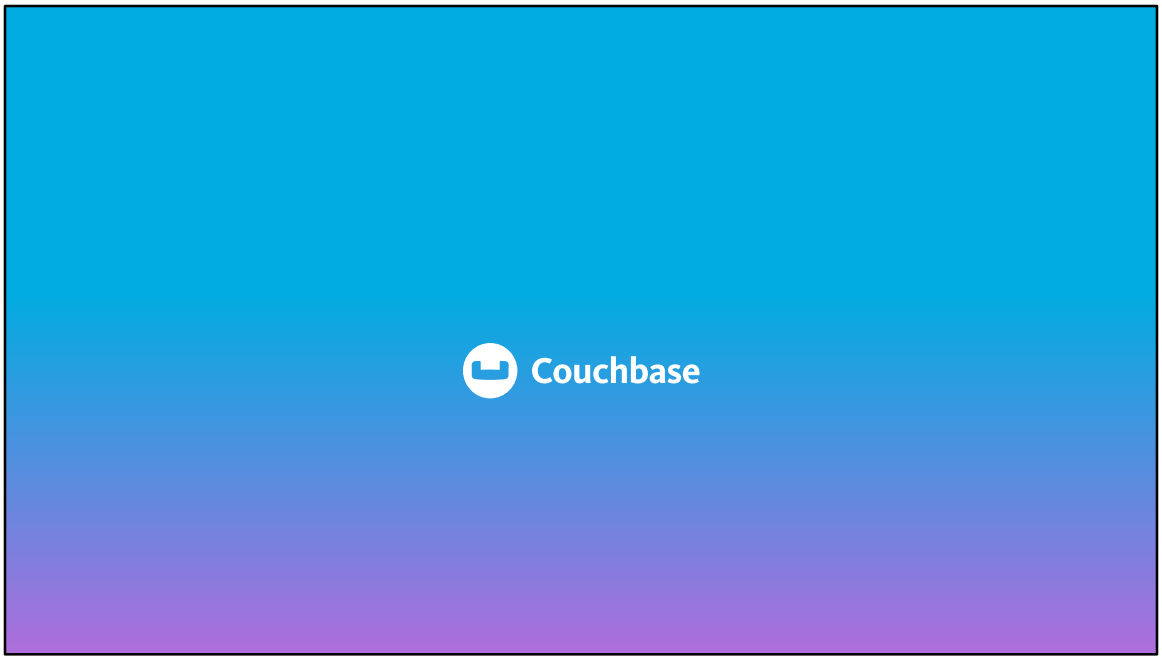
In some cases, it may not be desirable to log events incurred by particular users: for example, authentication performed by the Full Administrator. These users can be specified in the Ignore Filterable Events From These Users field. As the placeholder indicates, specification should take the form username/external or username/couchbase, according to the domain in which the user is registered. (See Authentication, for information on authentication domains.) Left-click on the Save button, to save the list of users.

Auditing a successful login



Here is an example of a successful login audit event,
It shows when did this event happen
What was the event, in this case its login
And who was the user that logged in, if you notice it's not CB Administrator login, but an LDAP user who is mapped to CB admin role.
Finally the remote IP and port details, to know from which machine the login was initiated.

For create bucket this output would be entirely different, it will give you the details like what the bucket name, and other relevant details about the bucket.


Couchbase