



# Manuale operativo MongoDB



---

# Contents

---

<b>I</b>	<b>Cluster MongoDB</b>	<b>3</b>
<b>1</b>	<b>Architettura</b>	<b>5</b>
1.1	Sharding . . . . .	5
1.2	Cluster . . . . .	5
1.3	Shard . . . . .	5
1.4	Config Servers . . . . .	6
1.5	Mongos . . . . .	6
1.6	Replica set . . . . .	6
1.7	Primary . . . . .	6
1.8	Secondary . . . . .	7
1.9	Arbiter . . . . .	7
1.10	Architettura di deploy . . . . .	7
<b>2</b>	<b>Aggiunta di un nodo ad un replica set</b>	<b>9</b>
2.1	Downlaod dei file necessari . . . . .	9
2.2	Restore dei dati . . . . .	9
2.3	Aggiunta al Replica Set . . . . .	10
<b>3</b>	<b>Inizializzazione di un Replica Set</b>	<b>11</b>
3.1	Downlaod dei file necessari . . . . .	11
3.2	Cambio dei parametri di configurazione . . . . .	11
3.3	Inizializzazione del Replica Set . . . . .	12
3.4	Aggiungere gli altri nodi . . . . .	12
<b>4</b>	<b>Aggiunta di una shard al cluster</b>	<b>13</b>
4.1	Collegamento a mongos . . . . .	13
4.2	Aggiunta della shard . . . . .	13
<b>5</b>	<b>Inizializzazione del cluster</b>	<b>15</b>



Il seguente manuale ha come scopo quello di guidare nella configurazione, nell'amministrazione e nella risoluzione dei problemi relativi al **Cluster MongoDB**, facente parte dello **Speed Layer** della **Smart Data Platform**. È diviso in 5 parti: **Cluster MongoDB**, Troubleshooting, Benchmark, Backup, Sicurezza; ogni parte è sua volta suddivisa in capitoli, che vanno ad illustrare nel dettaglio le operazioni da svolgere per la corretta amministrazione del **Cluster MongoDB**.



## Part I

# CLUSTER MONGODB

In questa parte del manuale è esplicita l'architettura del *Cluster* MongoDB, descrivendo i vari nodi, il loro funzionamento e l'attuale *Architettura di deploy* della **Smart Data Platform**.

Inoltre è spiegato come inizializzare un *Cluster* ex-novo, come aggiungere una *Shard* al *Cluster* e come aggiungere altri nodi ad un *Replica set*.





---

# Architettura

---

Il seguente capitolo è suddiviso in 2 parti, nella prima parte viene spiegato e definito brevemente il significato e il funzionamento di un *Cluster* MongoDB in *Sharding*. Sono presenti la descrizione e le definizioni dei vari elementi del cluster.

Nella seconda parte è illustrata l'*Architettura di deploy* del cluster specifica per la **Smart Data Platform**, con i nomi degli host, gli indirizzi e tutte le informazioni relative ai nodi che compongono il cluster.

## 1.1 Sharding

Lo sharding è un metodo di partizionamento dei dati all'interno diversi nodi detti *Shard*, permette di operare con grosse moli di dati ed avere performance che tramite un singolo nodo non sarebbe possibile raggiungere. Il partizionamento dei dati deve essere abilitato ed avviene per singola *collection*, la politica di partizionamento è dettata dalla scelta della *shard\_key*. In MongoDB lo sharding è molto flessibile e configurabile a seconda dell'esigenza, nel caso della **Smart Data Platform**, essendo il *Cluster* utilizzato per lo **Speed Layer** di inserimento dati, è stato configurato per favorire la velocità di inserimento dei dati, i dettagli della configurazione verranno forniti più avanti nella sezione relativa all'*Architettura di deploy*.

## 1.2 Cluster

Per cluster si intende l'insieme delle *Shard* (ognuna delle quali a sua volta è un *Replica set*) dai 3 *Config Servers* e dalle istanze *Mongos*.

## 1.3 Shard

Le *Shard* sono gli elementi del *Cluster* dove sono contenuti i dati partizionati, nel caso di *collection* non shardate i dati saranno residenti nella *Primary Shard*. Ciascuna *shard* è un *Replica set* a se stante e deve essere aggiunta al *Cluster* seguendo la procedura illustrata nel capitolo "*Aggiunta di una shard al cluster*". La lettura dei dati dal *Cluster* avviene tramite *Mongos*, collegandosi direttamente ad una *shard* tramite la shell *mongo* si leggeranno solo i dati partizionati presenti nella *shard* stessa.

---

**Note:** In ambiente di sviluppo è possibile utilizzare delle singole istanze *mongod* al posto dei *Replica set*.

---

### 1.3.1 Primary Shard

La Primary Shard si differenzia dalle altre *Shard* per il semplice fatto che contiene tutte le *collection* non shardate, non ha nulla a che fare con il nodo Primary del *Replica set*.

## 1.4 Config Servers

Un Config Server è un'istanza *mongod* configurata appositamente per svolgere tale ruolo. All'interno di un *Cluster* devono essere sempre presenti 3 Config Server che contengono i **metadati** del cluster all'interno del **database config**. I **metadati** contengono le informazioni necessarie ai *Mongos* per operare sui dati della *Shard* corretta.

Una *collection* shardata all'interno del *Cluster* viene suddivisa in *chunks* tra le varie *Shard* dal **bilanciamento**, le informazioni sulla posizione dei *chunks* all'interno del *Cluster* risiedono nel **database config**, ogni qualvolta avviene un **bilanciamento** dei *chunks* i **metadati** vengono aggiornati.

I 3 Config Server sono in **mirroring** (pur non essendo un *Replica set*) e devono sempre essere disponibili, nel caso in cui anche solo 1 dei Config Server non sia disponibile il database contenente i **metadati** del cluster andrà in modalità **sola lettura** e non avverranno più operazioni di bilanciamento dei *chunks*.

**Caution:** A partire dalla versione 3.2 di MongoDB i Config Server dovranno necessariamente far parte di un *Replica set* configurato appositamente per agire da Config Server, per questo motivo non sarà più necessario avere 3 istanze *mongod* in mirroring ma basterà configurare il *Replica set* dei Config Server.

## 1.5 Mongos

Il *mongos* è il servizio che si occupa di eseguire le operazioni di lettura e scrittura sul *Cluster*, si appoggia ai *Config Servers* per leggere i **metadati** del cluster e indirizzare alla *Shard* le operazioni, è consigliato l'utilizzo di 1 *mongos* per **application server** residente sull'application server stesso per assicurare che sia sempre attivo. Lato applicativo o per la shell *mongo* è trattato esattamente come una istanza *mongod*.

## 1.6 Replica set

Un Replica Set è un insieme di server *mongod* con i dati in replica, il numero di server può variare da 3 a 50 ma solo 7 saranno autorizzati al voto di elezione del *Primary*, un nodo all'interno del Replica Set può agire da *Primary*, *Secondary* o *Arbiter*

## 1.7 Primary

Solo un nodo all'interno del *Replica set* può essere **primary**. Come comportamento di default, tutte le operazioni di lettura e scrittura avvengono sul nodo **primary** e in seguito vengono replicate sui *Secondary* utilizzando l'*oplog*. All'interno del *Replica set* deve essere sempre presente un **primary**, nel momento in cui il **primary** non sarà più disponibile avverrà l'elezione del nuovo **primary** tra i *Secondary* che possono essere eletti.

## 1.8 Secondary

Tutti i nodi contenenti dati all'interno del *Replica set* sono considerati **secondary** (gli *Arbiter* non contengono dati). Sui nodi **secondary** non è possibile scrivere direttamente, i dati sono replicati dal *Primary* ripetendo le operazioni scritte sull'*oplog*. È invece possibile effettuare le operazioni di lettura per alleggerire il carico di lavoro del *Primary* ma si otterranno dei dati "vecchi" pari al tempo di replica.

## 1.9 Arbiter

Un nodo **arbiter** non contiene dati e agisce solo per la votazione del *Primary*, deve essere presente solo nel caso in cui nel *Replica set* ci sia il rischio di avere voti pari nel momento in cui si andrà a votare il nuovo *Primary*.

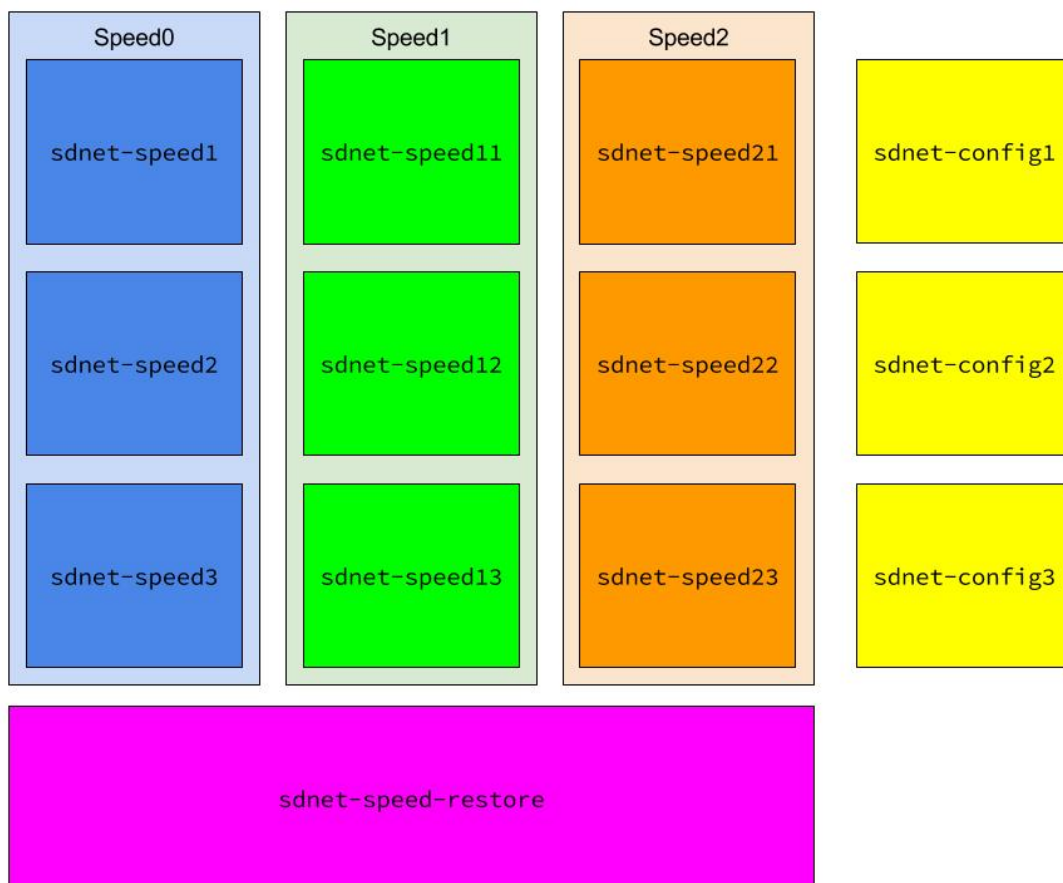
## 1.10 Architettura di deploy

Nello **Speed Layer** della **Smart Data Platform** è stato deciso di iniziare con un ambiente di produzione composto dai seguenti nodi:

- 3 *Shard*
- 3 *Config Servers*

Ciascuna *Shard* è 1 *Replica set* composto da:

- 3 *Primary/Secondary*
- 1 nodo non eleggibile a *Primary* utilizzato per il backup



I 3 config server sono i seguenti:

Nodo	Host	#CPU	RAM	Rete	Storage
sdnet-config1	sdnet-config1.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 30 GB Dati
sdnet-config2	sdnet-config2.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 30 GB Dati
sdnet-config3	sdnet-config3.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 30 GB Dati

Le 3 shard sono così suddivise:

<b>speed0</b>	Nodo	Host	#CPU	RAM	Rete	Storage
	sdnet-speed1	sdnet-speed1.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed2	sdnet-speed2.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed3	sdnet-speed3.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati

<b>speed1</b>	Nodo	Host	#CPU	RAM	Rete	Storage
	sdnet-speed11	sdnet-speed11.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed12	sdnet-speed12.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed13	sdnet-speed13.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati

<b>speed2</b>	Nodo	Host	#CPU	RAM	Rete	Storage
	sdnet-speed21	sdnet-speed21.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed22	sdnet-speed22.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati
	sdnet-speed23	sdnet-speed23.sdp.csi.it	1	4 GB	1 GB dati/1 GB backup	20 GB SO + 110 GB Dati

I 3 nodi di **backup** sono tutti sullo stesso host **sdnet-speed-restore** in modo da avere una singola macchina dalla quale gestire i backup delle 3 *Shard*:

Nodo	Host	#CPU	RAM	Rete	Storage
sdnet-speed-restore	sdnet-speed-restore.sdp.csi.it				

---

# Aggiunta di un nodo ad un replica set

---

Questo capitolo parla delle operazioni da effettuare per aggiungere un nodo ad un *Replica set* già esistente, è un'operazione che può essere eseguita "a caldo" senza necessità di dare disservizio.

Per comodità il nuovo nodo verrà chiamato `sdnet-speed4.sdp.csi.it` e sarà aggiunto al *Replica set* `speed0`. Nel caso in cui si voglia aggiungere un nodo con un host diverso, o nel caso in cui si voglia aggiungere il nodo ad un differente *Replica set* sarà necessario utilizzare i nomi corretti.

## 2.1 Downlaod dei file necessari

Collegarsi tramite ssh al nuovo nodo:

```
$ ssh sdnet-speed4.sdp.csi.it
```

Creare la cartella necessaria ad ospitare i file:

```
$ mkdir -p /data/mongodb/conf/
$ mkdir -p /data/mongodb/data/mongod-wiredTiger
...
```

Scaricare da un nodo già configurato del *Replica set* i seguenti file: `mongod` (script di avvio), `mongod.conf`, `keyfile`:

```
$ sudo scp sdnet-speed1.sdp.csi.it:/etc/init.d/mongod /etc/init.d/mongod
$ sudo scp sdnet-speed1.sdp.csi.it:/data/mongodb/conf/mongod.conf \
/data/mongodb/conf/mongod.conf
...
```

## 2.2 Restore dei dati

*Nel caso in cui si sta inizializzando un nuovo :ref:'cluster' da 0 questo paragrafo si può saltare.*

Per popolare il nuovo nodo con i dati esistono 2 possibilità:

- aggiungerlo al *Replica set* ed aspettare che venga eseguita la sincronizzazione da 0
- ripristinare i dati presi da un altro nodo

Vista la velocità e visto l'utilizzo di un nodo del *Replica set* utilizzato esclusivamente per il backup si è deciso di utilizzare la seconda possibilità.

Scaricare la cartella dei dati prendendola dal nodo di backup (sdnet-speed-restore):

```
$ sudo scp -r sdnet-speed-restore.sdp.csi.it:/backup_folder \
/data/mongodb/data/mongod-wiredTiger
```

Lanciare l'istanza mongod:

```
$ sudo /etc/init.d/mongod start
```

## 2.3 Aggiunta al Replica Set

Una volta lanciato il sever è necessario aggiungere al *Replica set* il nuovo nodo appena configurato, queste operazioni vanno eseguite sul *Primary* del *Replica set*.

Collegarsi al nodo sdnet-speed1.sdp.csi.it e verificare che sia *Primary*:

```
$ ssh sdnet-speed1.sdp.csi.it
$ mongo --port 27018
> use admin
> db.auth( |db_user| , |db_password| )
> db.isMaster()
```

Nel caso in cui sia *Secondary* provare con gli host sdnet-speed2.sdp.csi.it e sdnet-speed3.sdp.csi.it

Una volta collegati al corretto *Primary* aggiungere il nuovo nodo al *Replica set*:

```
> rs.add("sdnet-speed4.sdp.csi.it:27018")
```

Verificare che il nuovo nodo sia stato correttamente aggiunto:

```
> rs.conf()
```

Il JSON di output generato ha una chiave *members* in cui è presente la lista di tutti i nodi del *Replica set*, verificare che sia presente l'host del nuovo nodo appena aggiunto (sdnet-speed4.sdp.csi.it).

---

## Inizializzazione di un Replica Set

---

Questa procedura servirà ogniqualvolta si vorrà aggiungere una nuova *Shard* al *Cluster* in quanto una *Shard* deve essere necessariamente un *Replica set*.

Per comodità il nuovo *Replica set* verrà chiamato `speed3`, i relativo host che ne faranno parte saranno `sdnet-speed31.sdp.csi.it`, `sdnet-speed32.sdp.csi.it`, `sdnet-speed33.sdp.csi.it`. Nel caso in cui si voglia aggiungere una shard con un nome diverso, sarà necessario utilizzare i nomi corretti.

### 3.1 Downlaod dei file necessari

Collegarsi tramite ssh al nodo 1 del nuovo *Replica set*:

```
$ ssh sdnet-speed31.sdp.csi.it
```

Creare la cartella necessaria ad ospitare i file:

```
$ mkdir -p /data/mongodb/conf/  
$ mkdir -p /data/mongodb/data/mongod-wiredTiger  
...
```

Scaricare da un nodo già configurato del *Replica set* i seguenti file: `mongod` (script di avvio), `mongod.conf`, `keyfile`:

```
$ sudo scp sdnet-speed1.sdp.csi.it:/etc/init.d/mongod /etc/init.d/mongod  
$ sudo scp sdnet-speed1.sdp.csi.it:/data/mongodb/conf/mongod.conf \  
/data/mongodb/conf/mongod.conf  
...
```

### 3.2 Cambio dei parametri di configurazione

Nel file di configurazione `mongod.conf` appena scaricato andare a modificare il valore della chiave `replSet` mettendo il nome del nuovo *Replica set*:

```
replSet=speed0
```

### 3.3 Inizializzazione del Replica Set

Lanciare l'istanza mongod:

```
$ sudo /etc/init.d/mongod start
```

Collegarsi al mongod appena lanciato utilizzando la shell mongo:

```
$ mongo --port 27017
```

Inizializzare il *Replica set*:

```
> rs.initiate()
```

### 3.4 Aggiungere gli altri nodi

Per aggiungere gli altri nodi al *Replica set* seguire la procedura illustrata nel capitolo [Aggiunta di un nodo ad un replica set](#)



---

## Aggiunta di una shard al cluster

---

Una *Shard* deve essere obbligatoriamente un *Replica set*, nel caso in cui non sia già stato configurato un nuovo *Replica set*, è necessario procedere alla sua inizializzazione seguendo quanto spiegato nel capitolo *Inizializzazione di un Replica Set*.

L'operazione di aggiunta di una nuova *Shard* può essere eseguita "a caldo" senza necessità di dare disservizio, considerare comunque che sarà poi necessario che il *balancer* migri i *chunk* di dati e potrebbe causare un degrado del servizio.

Per comodità la *Shard* aggiunta verrà chiamata *speed3*, il relativo host scelto per fare l'aggiunta al *Cluster* sarà *speed3/sdnet-speed31.sdp.csi.it*. Nel caso in cui si voglia aggiungere una *Shard* con un nome diverso, sarà necessario utilizzare i nomi corretti.

### 4.1 Collegamento a mongos

Collegarsi tramite ssh ad un nodo avente un *Mongos* in esecuzione:

```
$ ssh sdnet-speed1.sdp.csi.it
```

Collegarsi al *Mongos* utilizzando la shell *mongo* e autenticarsi:

```
$ mongo --port 27019
> use admin
> db.auth( |db_user| , |db_password| )
```

### 4.2 Aggiunta della shard

Aggiungere la *Shard* al *Cluster*, specificando prima il nome del *Replica set* seguito dall'host di uno dei membri:

```
> sh.addShard( "speed3/sdnet-speed31.sdp.csi.it" )
```



---

# Inizializzazione del cluster

---

todo