

- `logAppend` is `true`, which ensures that `mongod` does not overwrite an existing log file following the server start operation.
- `storage.journal.enabled` is `true`, which enables *journaling*. Journaling ensures single instance write-durability. 64-bit builds of `mongod` enable journaling by default. Thus, this setting may be redundant.

Given the default configuration, some of these values may be redundant. However, in many situations explicitly stating the configuration increases overall system intelligibility.

### Security Considerations

The following collection of configuration options are useful for limiting access to a `mongod` instance. Consider the following settings, shown in both YAML and older configuration file format:

In YAML format

```
security:
  authorization: enabled
net:
  bindIp: 127.0.0.1,10.8.0.10,192.168.4.24
```

Or, if using the older [older configuration file format](#)<sup>46</sup>:

```
bind_ip = 127.0.0.1,10.8.0.10,192.168.4.24
auth = true
```

Consider the following explanation for these configuration decisions:

- “`bindIp`” has three values: `127.0.0.1`, the localhost interface; `10.8.0.10`, a private IP address typically used for local networks and VPN interfaces; and `192.168.4.24`, a private network interface typically used for local networks.

Because production MongoDB instances need to be accessible from multiple database servers, it is important to bind MongoDB to multiple interfaces that are accessible from your application servers. At the same time it’s important to limit these interfaces to interfaces controlled and protected at the network layer.

- “`authorization`” is `true` enables the authorization system within MongoDB. If enabled you will need to log in by connecting over the `localhost` interface for the first time to create user credentials.

**See also:**

[Security Concepts](#) (page 320)

### Replication and Sharding Configuration

**Replication Configuration** *Replica set* configuration is straightforward, and only requires that the `replSetName` have a value that is consistent among all members of the set. Consider the following:

In YAML format

```
replication:
  replSetName: set0
```

Or, if using the [older configuration file format](#)<sup>47</sup>:

```
replSet = set0
```

---

<sup>46</sup><http://docs.mongodb.org/v2.4/reference/configuration-options>

<sup>47</sup><http://docs.mongodb.org/v2.4/reference/configuration-options>