Node settings

Self-Managed

Any time that you start an instance of Elasticsearch, you are starting a *node*. A collection of connected nodes is called a cluster. If you are running a single node of Elasticsearch, then you have a cluster of one node.
Every node in the cluster can handle HTTP and transport traffic by default. The transport layer is used exclusively for communication between nodes; the HTTP layer is used by REST clients.
All nodes know about all the other nodes in the cluster and can forward client requests to the appropriate node.
Tip
The performance of an Elasticsearch node is often limited by the performance of the underlying storage. Review our recommendations for optimizing your storage for indexing and search.

Node name setting

Elasticsearch uses `node.name` as a human-readable identifier for a particular instance of Elasticsearch. This name is included in the response of many APIs. The node name defaults to the hostname of the machine when Elasticsearch starts, but can be configured explicitly in `elasticsearch.yml`:

```
node.name: prod-data-2
```

Node role settings

You define a node's roles by setting `node.roles` in `elasticsearch.yml`. If you set `node.roles`, the node is only assigned the roles you specify. If you don't set `node.roles`, the node is assigned the following roles:

- `master`
- `data`
- `data_content`
- `data_hot`
- `data_warm`
- `data_cold`
- `data_frozen`
- `ingest`

- `ml`
- `remote_cluster_client`
- `transform`

The following additional roles are available:

- `voting_only`

If If you set `node.roles` to an empty array (`node.roles: [ ]`), then the node is considered to be a coordinating only node.

Important

If you set `node.roles`, ensure you specify every node role your cluster needs. Every cluster requires the following node roles:

- `master`
- `data_content` and `data_hot` OR `data`

Some Elastic Stack features also require specific node roles:

- Cross-cluster search and cross-cluster replication require the `remote_cluster_client` role.
- Stack Monitoring and ingest pipelines require the `ingest` role.
- Fleet, the Elastic Security app, and transforms require the `transform` role. The `remote_cluster_client` role is also required to use cross-cluster search with these features.
- Machine learning features, such as anomaly detection, require the `ml` role.

As the cluster grows and in particular if you have large machine learning jobs or continuous transforms, consider separating dedicated master-eligible nodes from dedicated data nodes, machine learning nodes, and transform nodes.

To learn more about the available node roles, see *Node roles*.

Node data path settings

`path.data`

Every data and master-eligible node requires access to a data directory where shards and index and cluster metadata will be stored. The `path.data` defaults to `$ES_HOME/data` but can be configured in the `elasticsearch.yml` config file an absolute path or a path relative to `$ES_HOME` as follows:

```
path.data:  /var/elasticsearch/data
```

Like all node settings, it can also be specified on the command line as:

```
./bin/elasticsearch -Epath.data=/var/elasticsearch/data
```

The contents of the `path.data` directory must persist across restarts, because this is where your data is stored. Elasticsearch requires the filesystem to act as if it were backed by a local disk, but this means that it will work correctly on properly-configured remote block devices (e.g. a SAN) and remote filesystems (e.g. NFS) as long as the remote storage behaves no differently from local storage. You can run multiple Elasticsearch nodes on the same filesystem, but each Elasticsearch node must have its own data path.

Tip

When using the `.zip` or `.tar.gz` distributions, the `path.data` setting should be configured to locate the data directory outside the Elasticsearch home directory, so that the home directory can be deleted without deleting your data! The RPM and Debian distributions do this for you already.

Warning

Don't modify anything within the data directory or run processes that might interfere with its contents. If something other than Elasticsearch modifies the contents of the data directory, then Elasticsearch may fail, reporting corruption or other data inconsistencies, or may appear to work correctly having silently lost some of your data. Don't attempt to take filesystem backups of the data directory; there is no supported way to restore such a backup. Instead, use Snapshot and restore to take backups safely. Don't run virus scanners on the data directory. A virus scanner can prevent Elasticsearch from working correctly and may modify the contents of the data directory. The data directory contains no executables so a virus scan will only find false positives.

Custom node attributes

If needed, you can add custom attributes to a node. These attributes can be used to filter which nodes a shard can be allocated to, or to group nodes together for shard allocation awareness.

Tip

You can also set a node attribute using the `-E` command line argument when you start a node:

```
./bin/elasticsearch -Enode.attr.rack_id=rack_one
```

`node.attr.<attribute-name>`

(Dynamic) A custom attribute that you can assign to a node. For example, you might assign a `rack_id` attribute to each node to ensure that primary and replica shards are not allocated on the same rack. You can specify multiple attributes as a comma-separated list.

Other node settings

More node settings can be found in *Configuring Elasticsearch* and Important Elasticsearch configuration, including:

- `cluster.name`
- `node.name`
- network settings