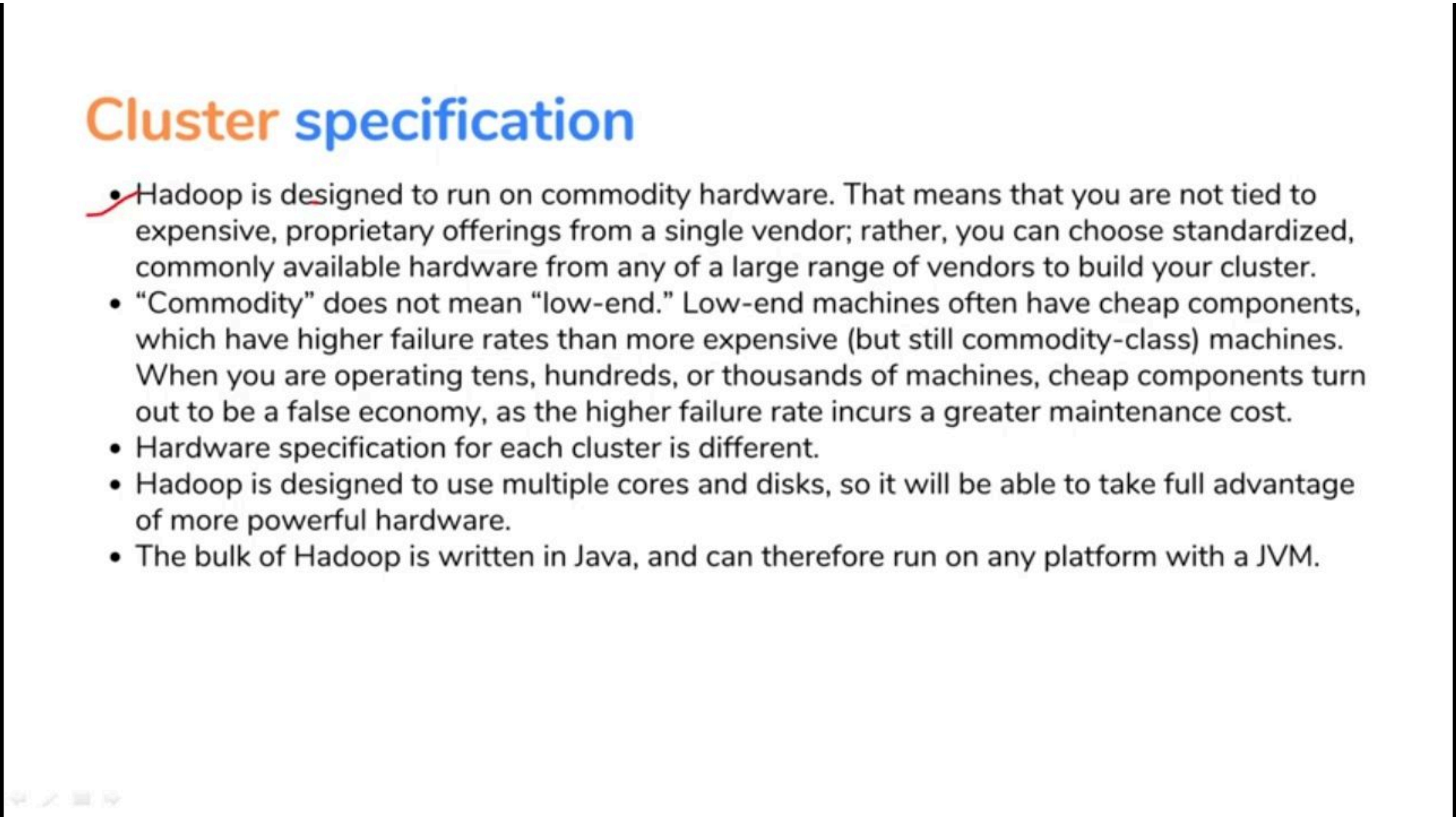
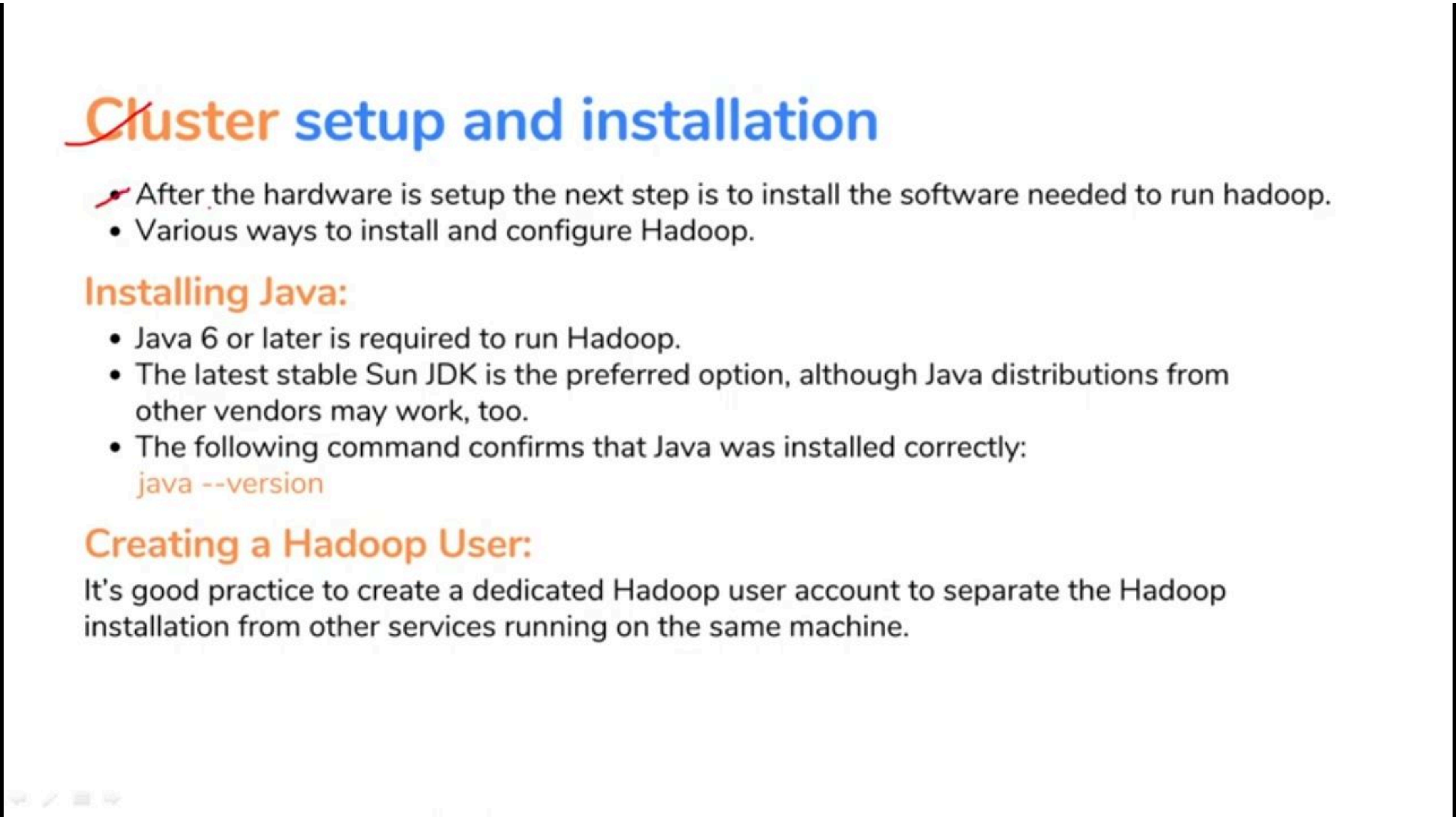


Cluster specification

- Hadoop is designed to run on commodity hardware. That means that you are not tied to expensive, proprietary offerings from a single vendor; rather, you can choose standardized, commonly available hardware from any of a large range of vendors to build your cluster.
- “Commodity” does not mean “low-end.” Low-end machines often have cheap components, which have higher failure rates than more expensive (but still commodity-class) machines. When you are operating tens, hundreds, or thousands of machines, cheap components turn out to be a false economy, as the higher failure rate incurs a greater maintenance cost.
- Hardware specification for each cluster is different.
- Hadoop is designed to use multiple cores and disks, so it will be able to take full advantage of more powerful hardware.
- The bulk of Hadoop is written in Java, and can therefore run on any platform with a JVM.



Cluster setup and installation

- After the hardware is setup the next step is to install the software needed to run hadoop.
- Various ways to install and configure Hadoop.

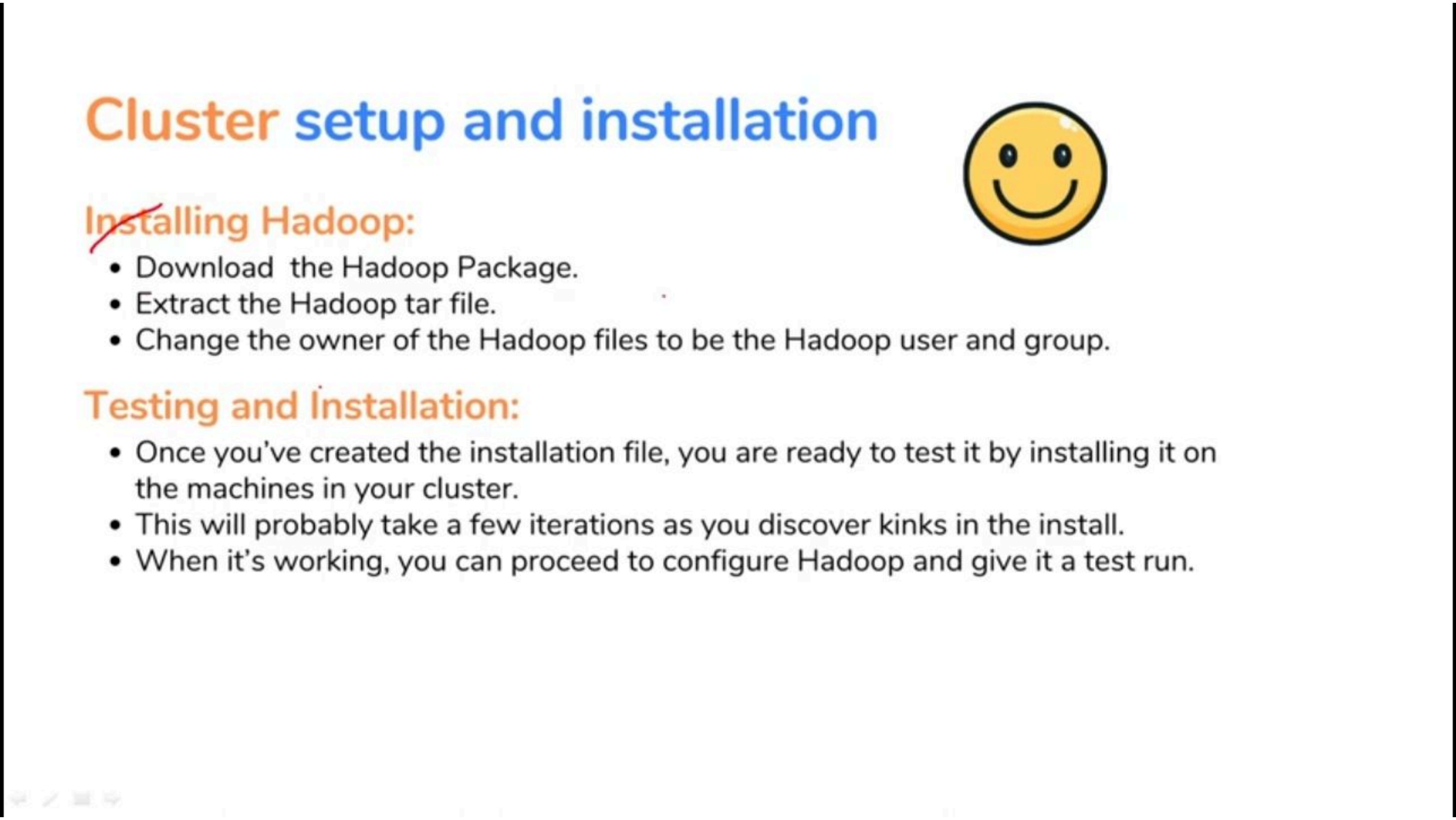
Installing Java:

- Java 6 or later is required to run Hadoop.
- The latest stable Sun JDK is the preferred option, although Java distributions from other vendors may work, too.
- The following command confirms that Java was installed correctly:

```
java --version
```

Creating a Hadoop User:

It's good practice to create a dedicated Hadoop user account to separate the Hadoop installation from other services running on the same machine.



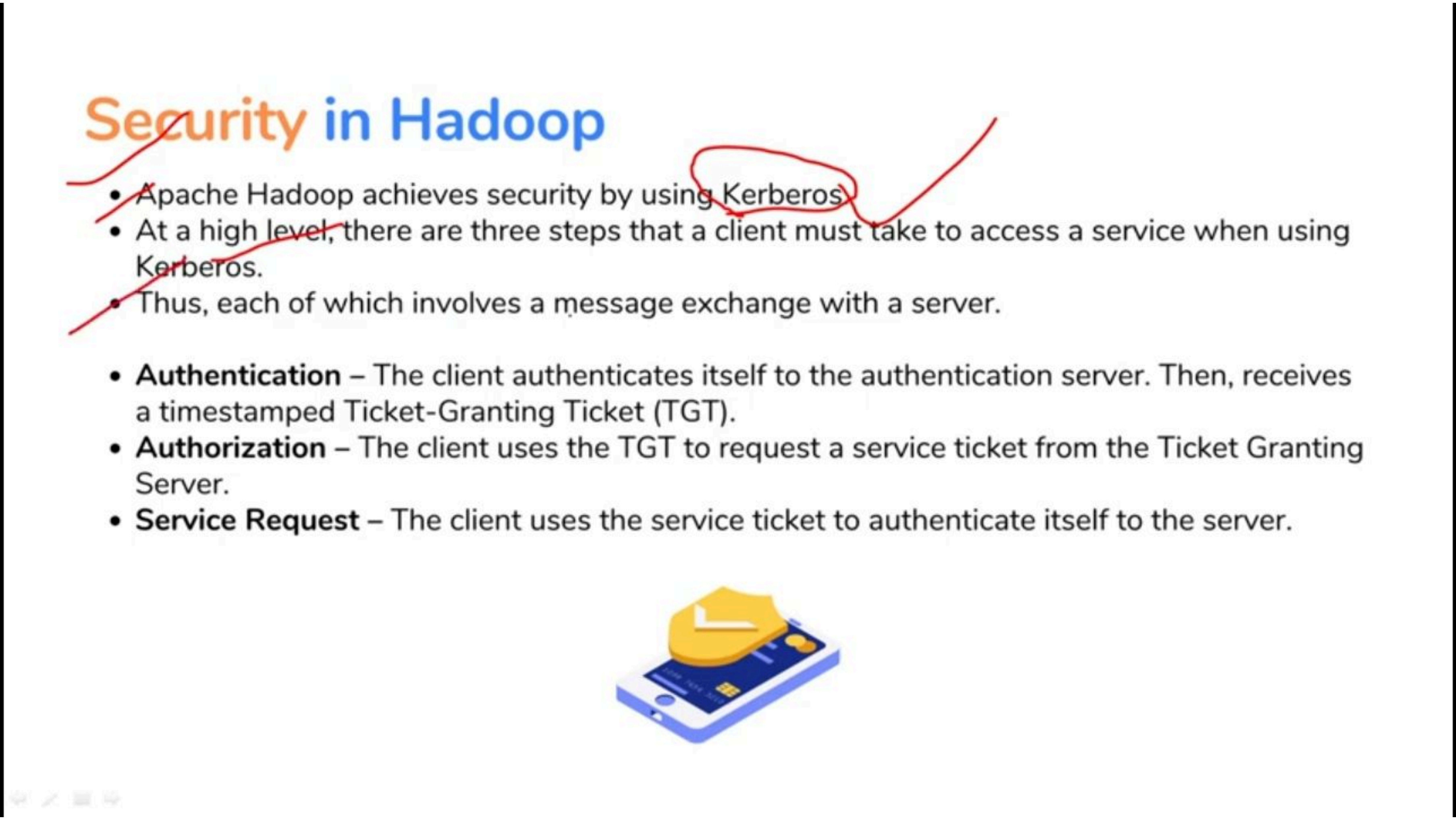
Cluster setup and installation

Installing Hadoop:

- Download the Hadoop Package.
- Extract the Hadoop tar file.
- Change the owner of the Hadoop files to be the Hadoop user and group.

Testing and Installation:

- Once you've created the installation file, you are ready to test it by installing it on the machines in your cluster.
- This will probably take a few iterations as you discover kinks in the install.
- When it's working, you can proceed to configure Hadoop and give it a test run.



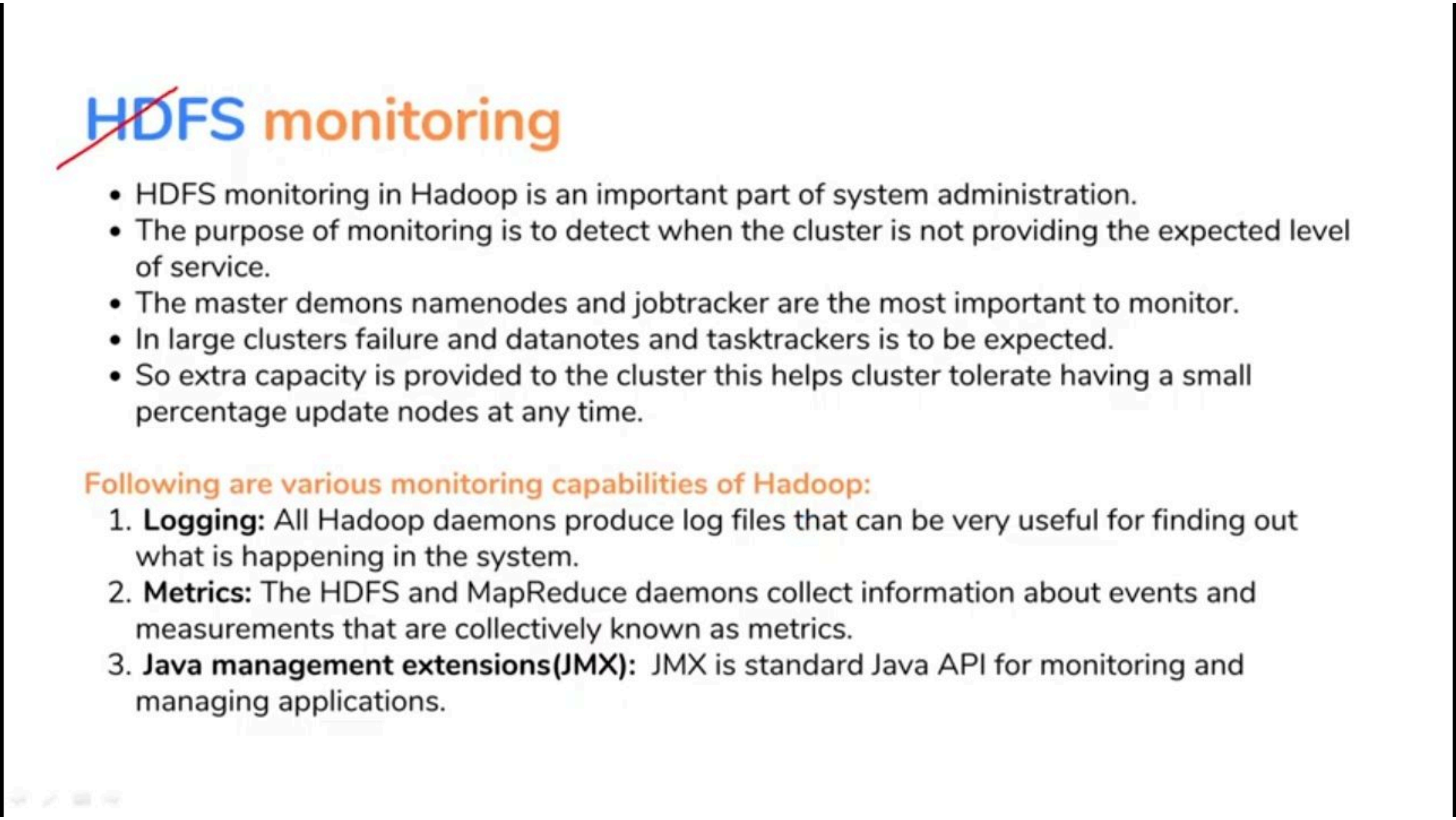
Security in Hadoop

- Apache Hadoop achieves security by using Kerberos.
- At a high level, there are three steps that a client must take to access a service when using Kerberos.
- Thus, each of which involves a message exchange with a server.

- **Authentication** – The client authenticates itself to the authentication server. Then, receives a timestamped Ticket-Granting Ticket (TGT).
- **Authorization** – The client uses the TGT to request a service ticket from the Ticket Granting Server.
- **Service Request** – The client uses the service ticket to authenticate itself to the server.

Administering Hadoop

- The person who administers Hadoop is called HADOOP ADMINISTRATOR.
- Some of the common administering tasks in Hadoop are :
- Monitor health of a cluster
- Add new data nodes as needed
- Optionally turn on security
- Optionally turn on encryption
- Recommended, but optional, to turn on high availability
- Optional to turn on MapReduce Job History Tracking Server
- Fix corrupt data blocks when necessary
- Tune performance

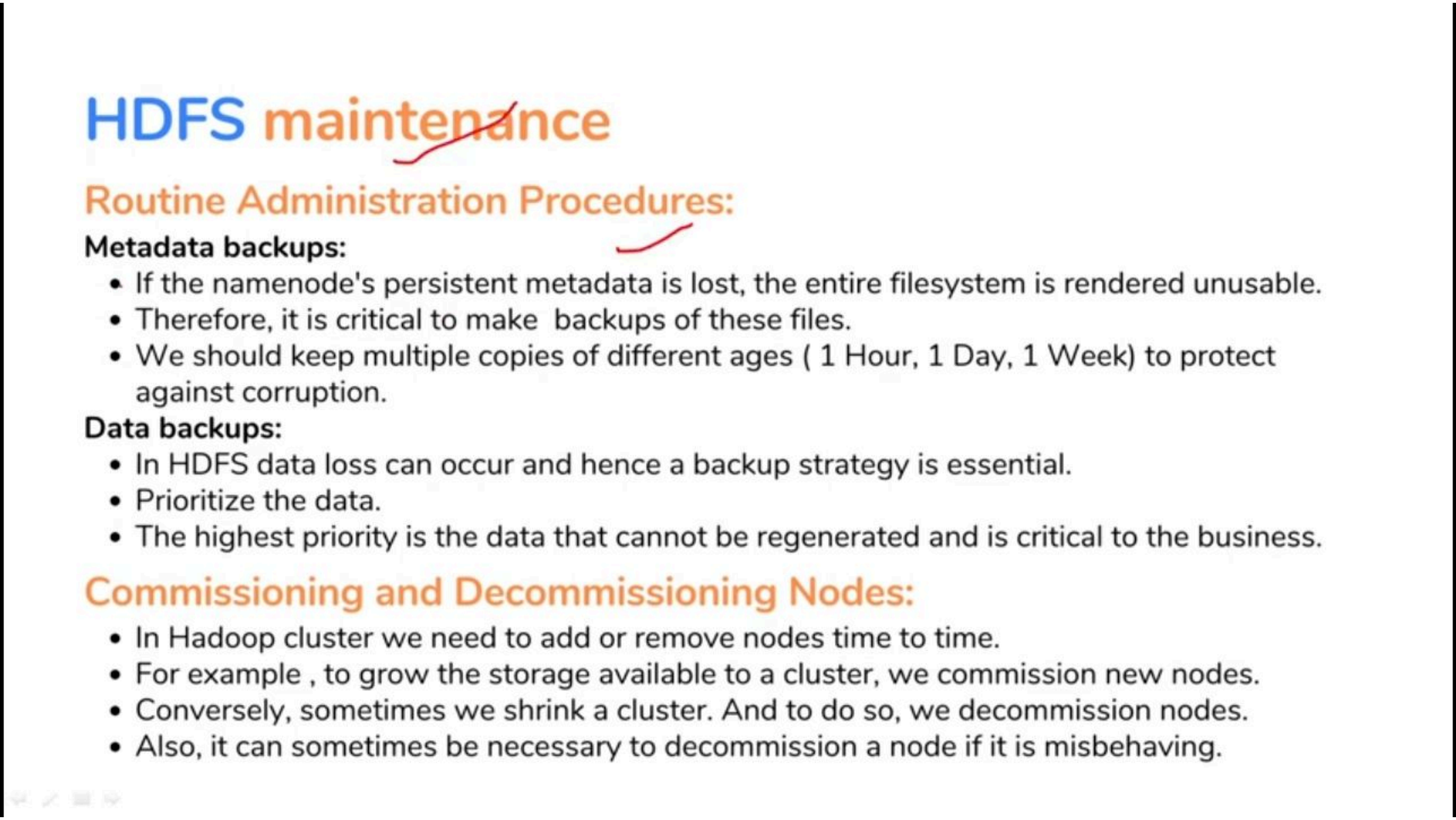


~~HD~~FS monitoring

- HDFS monitoring in Hadoop is an important part of system administration.
- The purpose of monitoring is to detect when the cluster is not providing the expected level of service.
- The master demons namenodes and jobtracker are the most important to monitor.
- In large clusters failure and datanotes and tasktrackers is to be expected.
- So extra capacity is provided to the cluster this helps cluster tolerate having a small percentage update nodes at any time.

Following are various monitoring capabilities of Hadoop:

1. **Logging:** All Hadoop daemons produce log files that can be very useful for finding out what is happening in the system.
2. **Metrics:** The HDFS and MapReduce daemons collect information about events and measurements that are collectively known as metrics.
3. **Java management extensions(JMX):** JMX is standard Java API for monitoring and managing applications.



HDFS maintenance

Routine Administration Procedures:

Metadata backups:

- If the namenode's persistent metadata is lost, the entire filesystem is rendered unusable.
- Therefore, it is critical to make backups of these files.
- We should keep multiple copies of different ages (1 Hour, 1 Day, 1 Week) to protect against corruption.

Data backups:

- In HDFS data loss can occur and hence a backup strategy is essential.
- Prioritize the data.
- The highest priority is the data that cannot be regenerated and is critical to the business.

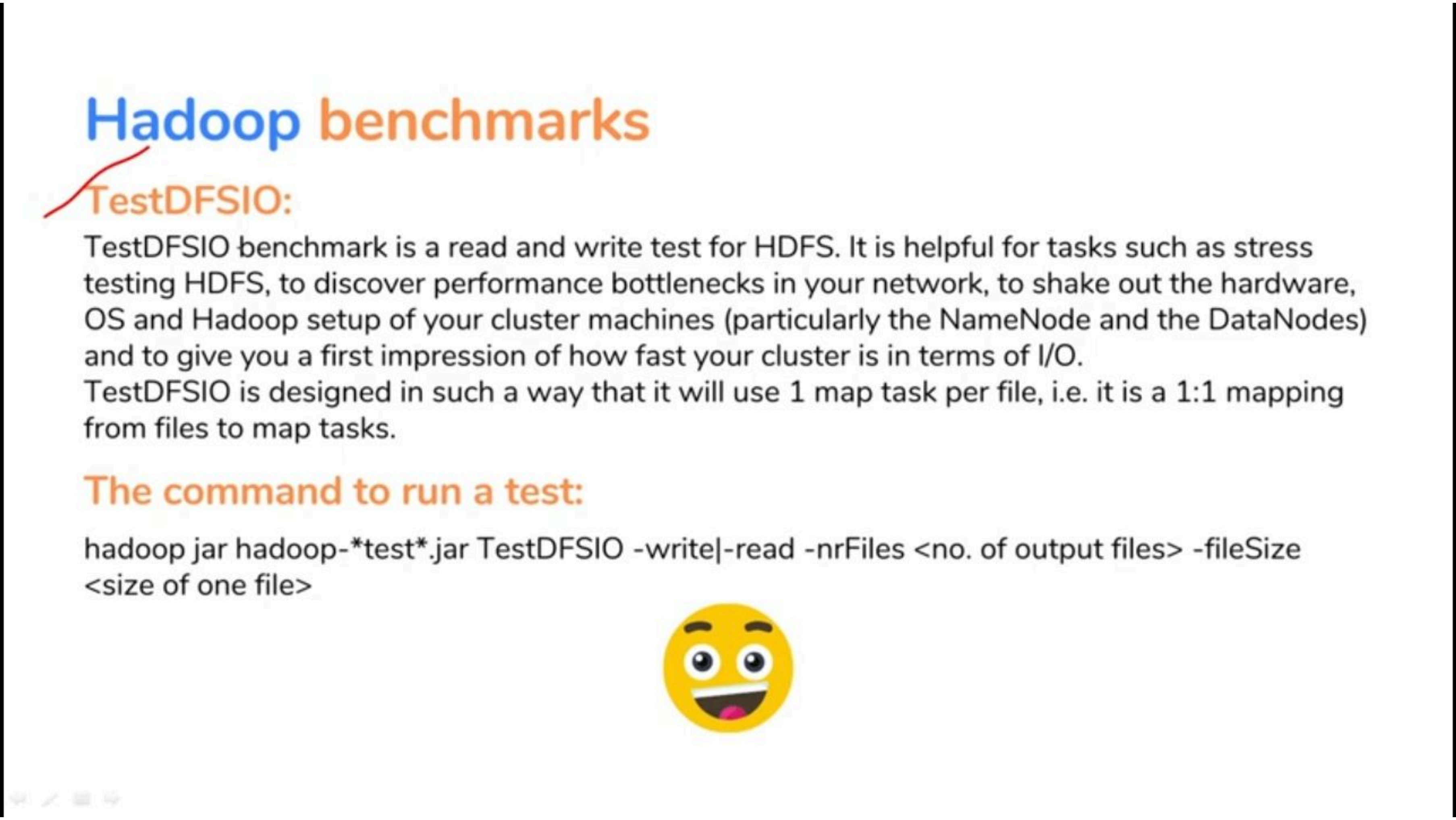
Commissioning and Decommissioning Nodes:

- In Hadoop cluster we need to add or remove nodes time to time.
- For example , to grow the storage available to a cluster, we commission new nodes.
- Conversely, sometimes we shrink a cluster. And to do so, we decommission nodes.
- Also, it can sometimes be necessary to decommission a node if it is misbehaving.

HDFS maintenance

Routine Administration Procedures:

- Upgrading an HDFS and MapReduce cluster requires careful planning.
- Part of the planning process should include a trial run on a small test cluster with a copy of data that you can afford to lose.



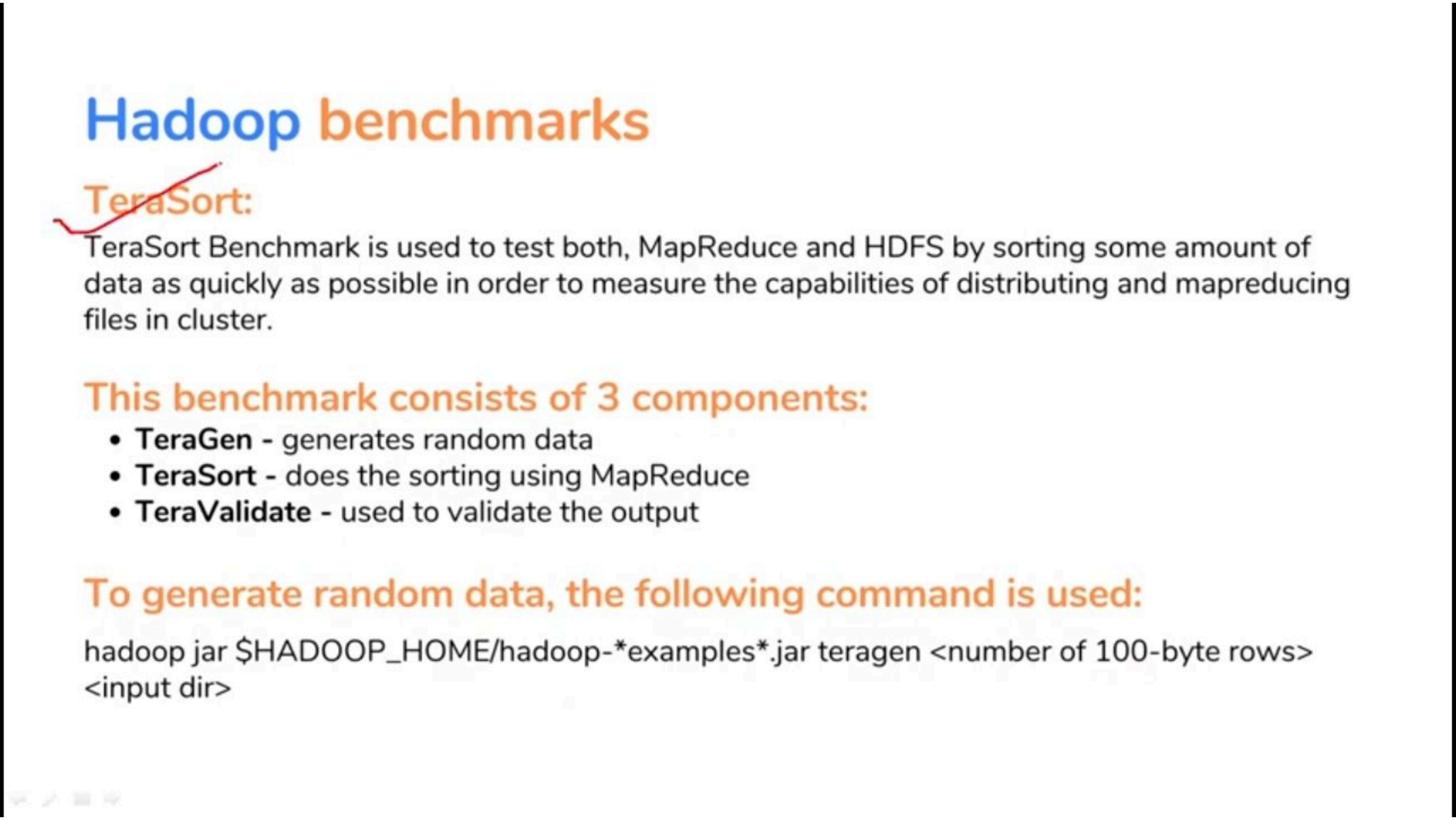
Hadoop benchmarks

TestDFSIO:

TestDFSIO benchmark is a read and write test for HDFS. It is helpful for tasks such as stress testing HDFS, to discover performance bottlenecks in your network, to shake out the hardware, OS and Hadoop setup of your cluster machines (particularly the NameNode and the DataNodes) and to give you a first impression of how fast your cluster is in terms of I/O.
TestDFSIO is designed in such a way that it will use 1 map task per file, i.e. it is a 1:1 mapping from files to map tasks.

The command to run a test:

```
hadoop jar hadoop-*test*.jar TestDFSIO -write|-read -nrFiles <no. of output files> -fileSize
<size of one file>
```



Hadoop benchmarks

TeraSort:

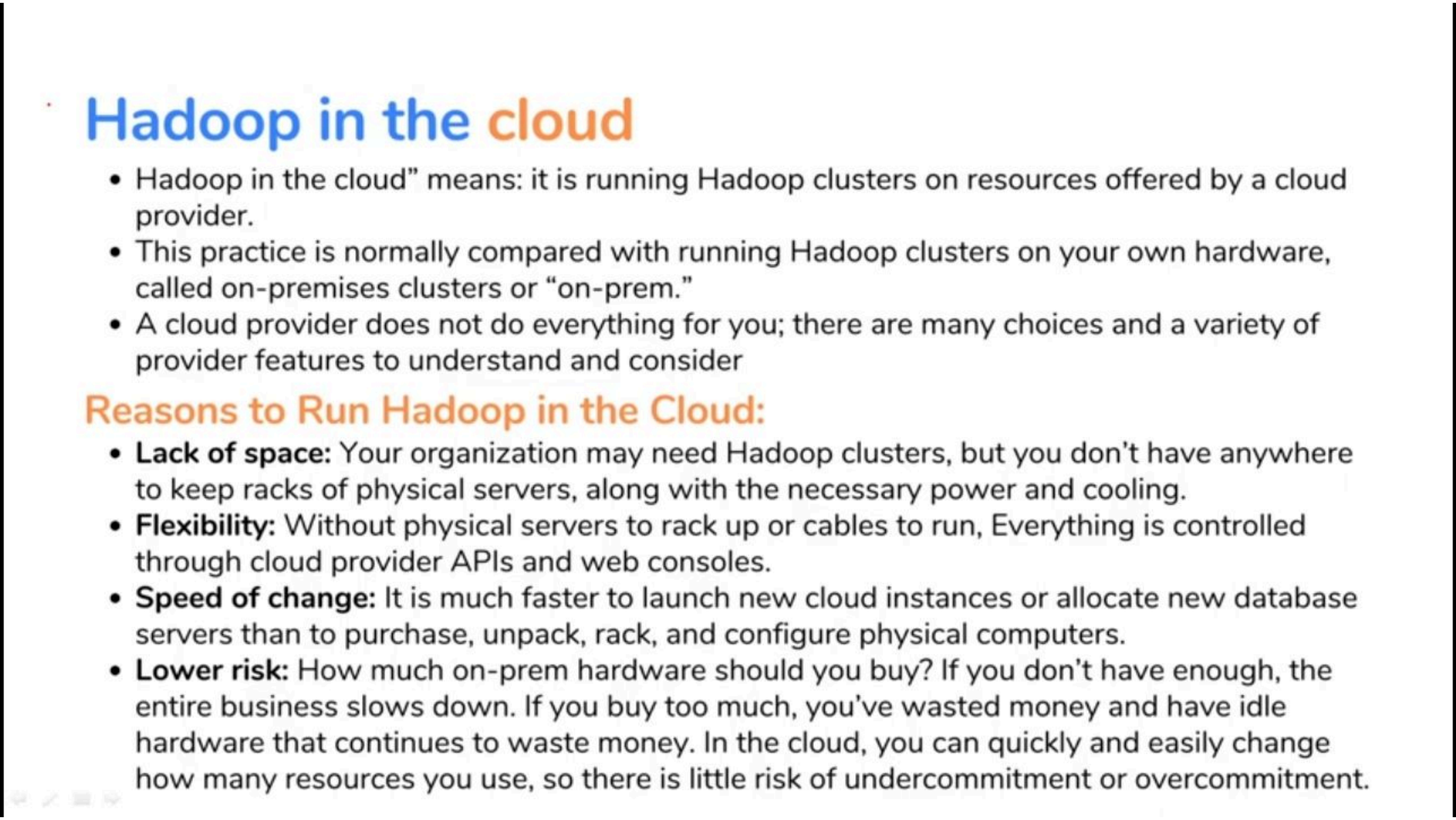
TeraSort Benchmark is used to test both, MapReduce and HDFS by sorting some amount of data as quickly as possible in order to measure the capabilities of distributing and mapreducing files in cluster.

This benchmark consists of 3 components:

- **TeraGen -** generates random data
- **TeraSort -** does the sorting using MapReduce
- **TeraValidate -** used to validate the output

To generate random data, the following command is used:

```
hadoop jar $HADOOP_HOME/hadoop-*examples*.jar teragen <number of 100-byte rows>
<input dir>
```



Hadoop in the cloud

- Hadoop in the cloud" means: it is running Hadoop clusters on resources offered by a cloud provider.
- This practice is normally compared with running Hadoop clusters on your own hardware, called on-premises clusters or "on-prem."
- A cloud provider does not do everything for you; there are many choices and a variety of provider features to understand and consider

Reasons to Run Hadoop in the Cloud:

- **Lack of space:** Your organization may need Hadoop clusters, but you don't have anywhere to keep racks of physical servers, along with the necessary power and cooling.
- **Flexibility:** Without physical servers to rack up or cables to run, Everything is controlled through cloud provider APIs and web consoles.
- **Speed of change:** It is much faster to launch new cloud instances or allocate new database servers than to purchase, unpack, rack, and configure physical computers.
- **Lower risk:** How much on-prem hardware should you buy? If you don't have enough, the entire business slows down. If you buy too much, you've wasted money and have idle hardware that continues to waste money. In the cloud, you can quickly and easily change how many resources you use, so there is little risk of undercommitment or overcommitment.

Hadoop in the cloud

- **Focus:** An organization using a cloud provider to rent resources, instead of spending time and effort on the logistics of purchasing and maintaining its own physical hardware and networks, is free to focus on its core competencies, like using Hadoop clusters to carry out its business. This is a compelling advantage for a tech startup.
- **Worldwide availability**
- **Capacity**

Reasons to Not Run Hadoop in the Cloud:

- **Simplicity**
- **High levels of control**
- **Unique hardware needs**
- **Saving money**

EIOV