1. 
2. Following is the quick overview of what we will be covering.
3. Text

   Description automatically generated with medium confidence
   1. What is **Hazelcast** and what is its **usages**?
   2. What are the **primary Data Structures in Hazelcast** & basic use cases of it?
   3. **Efficient Use of the Network**: As Hazelcast is distributed technology. So efficient use of network is important.
   4. **Monitoring**: How to monitor our Hazelcast Cluster.
4. **So, what is Hazelcast?**
   1. **Caching**:
      1. Firstly, it is **cache technology**. That is a **distributed cache**. ( Simply, it means to say that it is **Distributed Cache Technology**)
      2. Every client has access to the **cache data**. No matter where they reside.  
         The client could be on a different server (in the same rack) or on a different rack in a data center or in a different data center somewhere in another country.  
         A Data Center = A set of Racks.  
         Rack = A set of Servers.  
         Data Centers are set up distributed in different countries geographically.
   2. **Clustering**:
      1. Hazelcast is a **clustering technology** which means we can easily **scale out** to meet the demand **by starting off additional nodes**, which will **automatically join** the cluster.
   3. **No-SQL Key/Value Store**:
      1. Hazelcast can be used **as a in-memory No-SQL key/value data source**.
   4. **Application Scaling:**
      1. Hazelcast allows us to easily scale out our application by adding new nodes.
      2. The new nodes make the memory & CPU resources available to the **Hazelcast** cluster for utilization.
   5. **Messaging**:
      1. **Hazelcast** has a **Messing System** built into it which supports **publish/subscribe** & **producer/consumer models** all in-memory so super-fast.
   6. **Data Grid**:
      1. All of these features put together make it useful as a **Data Grid**.
      2. This is what allows us to create dynamic, fast, scalable & highly available & performant app.
         1. Dynamic: Because of Cluster.
         2. Fast: Because of In-Memory
         3. Highly Available: Because of backup copies.
         4. Performant: Because of in-memory.
5. **Who is using Hazelcast**?
   1. 
   2. As you can see the logos from the above snapshot, quit a number of well-known companies & various sectors.
   3. **Example**:
      1. America Express in Financial Industry.
      2. Cisco in Telecommunication.
      3. Companies offering ecommerce such as Domino, Target.
6. **Hazelcast is available in two editions**.
   1. **A standard Open-Source Edition**:
      1. 
      2. This edition is generally referred to as Hazelcast in their Documentation.
      3. It covers a lot of functionality & generally more than enough to get you up and running with Hazelcast.
      4. The source code is available on Github.   
         So, if you need to dive into a class provided by Hazelcast to understand something a bit better, you can easily do that.
   2. **Commercially Licensed Enterprise Edition**:
      1. Background pattern

         Description automatically generated with low confidence
      2. This edition is generally referred to as **Enterprise Hazelcast** in their Documentation.
      3. It is available under a commercial license & requires a payment.
      4. This edition provides some additional features which may be required for an enterprise app such as **Security**.
      5. It also provides the ability to synchronize two clusters which are generally distributed geographically using WAN Replication.
      6. Another feature is to cluster the session data from your Tomcat Web Server.  
         So, you can easily create a cluster of these Tomcat Servers. No matter which one your request hits, your session info will be available on it.  
         So, let’s look at hazel cast cluster.
7. **Hazelcast Cluster**:  
   Diagram

   Description automatically generated
   1. It is made of nodes and clients.
   2. Nodes represented within the cloud in the diagram make their(nodes) memory & CPU resources available to the cluster.
   3. Clients (which could be another server process, Desktop App or Mobile App) can access the cluster to use it as they wish.
   4. Clients are not members of the cluster. They can connect to the cluster but don’t provide memory & CPU resources to the cluster.
8. As this is get going type of course, we will be covering the topics that will get you up and running with Hazelcast.  
   We will not be covering all the Hazelcast Data Structures and distributed execution possibilities.   
   The idea being to give you enough information to get started and then you can build on your knowledge with the Hazelcast Documentation or other courses.
9. Text

   Description automatically generated