So let's start with the overview.

First let's have a look at the Supported node configurations.

A cluster can be as small as one single node. Running a single node cluster obviously means that if you lose the node, you will lose access to your data.

To be redundant at the node level, you can add a node to the existing cluster to form an HA-Pair.
A node in an HA-Pair shares the disks with the other node in the same HA-Pair.
If one node fails, the other node can takeover the aggregates of the failed node.

If you run an HA-Pair you will have something which is called the HA-Interconnect.
The HA-Interconnect is used to mirror the NVRAM between the two nodes. So that if a node fails, the data that was not yes written to disk is also available on the other node. After having taken over the aggregates of the failed node, the data will be written to disk.
Also, with an HA-Pair you will have a network which is called the Cluster Interconnect.
The cluster interconnect is used for heartbeat, configuration management and volumedata between the nodes of the cluster.

A cluster can be scaled out by adding more HA-Pairs to the cluster. Each HA-Pair will have to be connected to the ClusterInterconnect, and each HA-Pair will have its own HA-Interconnect.

Depending on the protocols you use a cluster can have as many as 24 nodes.
The maximum number of HA-PAIRS in a SAN environment depends on the controller-types you use.
Currently, with AllFlash arrays, you can have a maximum of 6 HA-Pairs which is 12 nodes, in a SAN environment.

The configuration limits of all sorts can be found at the hardware universe: hwu.netapp.com.

So, we can have a single node with one or more diskshelves, you can have an HA-Pair with one or more diskshelves. Or you can have a number of HA-Pairs.

Another important feature of the ONTAP cluster configuration is Storage Virtual Machines.

A cluster will always run Storage Virtual Machines. SVMs are logical representations of one of four types. We have a system type. There is only one system type SVM, this SVM is used to manage the cluster interconnect interfaces. We have the node type each node in your cluster is represented as a node svm. The admin type is the management svm. You will usually connect to this particular SVM if you want to manage your cluster. So the admin type svm basically represents your cluster.

And finally we have the data type. We can have up to 128 datatype SVMS in our cluster. This type of SVM is used to present data to our NAS and SAN clients.

DataSVMS are the svms that service the data. The data of an SVM is stored in volumes.
The volumes of an SVM are part of the namespace of the svm. Every svm has its own namespace.
So in short: the namespace of an SVM contains all the datavolumes.

A dataSVM will always have one and exactly one rootvolume. This is a small volume that is commonly used
to mount the datavolumes to. These mountpoints are called junction-paths. In a NAS environment the volumes will have to be mounted, otherwise they will not be accessible by clients.

The rootvolume of an SVM will should never contain data. The only thing you will find in the SVM's rootvolume are the junctionpaths of mounted data volumes.

The Cluster configuration.

The cluster configuration is stored in Replicated Databases. These replicated databases are located in the rootvolume of every node. These databases should always be in sync in the cluster. So this means that all nodes have the exact same configuration information about the cluster resources like volumes, ip-addresses etcetera.

Important to know that the rootvolume is mounted to the /mroot directory on the node.

Before we have a look at the command-line let's have a look at what these RDBS really are. There is five of them, and they all contain configuration information. The VLDB contains metainformation about volumes and aggregates, BCOMD has SAN info, think of igroups and such, VIFMGR has to do with logical interfaces, CRS is for replicating SVM configuration to other clusters and Mgmt is for all the thing that are not stored in one of the previous four. You should think of users, policies etcetera.

---
demo

So let's have a quick look at what we just discussed.
First we login to our cluster. The ip address via which I enter is 192.168.4.100.
I login as admin and view the nodes in my cluster.

So I run cluster show and I see that I have 4 nodes in my cluster. Because these are simulators we can't say that they are part of an HA-pair. Simulators unfortunately are in a non-shared environment, so each svm has its own disks. In real life, this four node cluster would consist of 2 ha-pairs.

Let's check the SVMs.
We run vserver show and see that we have 7 svms, 4 nodesvms, one admin svm and 2 datasvms
No if we go to the advanced mode and run the same command again, we see yet another svm that is of the type system. This one manages the cluster interconnects. We will have a look at that after we have discussed the networking overview later.

To see the volumes we run vol show and we see that each node has its own vol0. also each vol0 has its own aggregate in which it lives. The datavolumes are in data aggregates. we have n1_aggr1

and n2_aggr1 .
Both SVMs have 1 rootvolume and 1 datavolume each.

Networking obviously is a very important item in the clusterconfiguration.
First of all, if you run one or more HA-Pairs, there is the cluster interconnect. Each node will have a minimum of 2 10Gb interfaces that will have a single ip-address per port that is connected to the cluster interconnect network. These ports are configured during the cluster initialization or when a node joins a cluster.

If needed, you can add more interfaces to the cluster interconnect to grow the bandwidth.

Next to the cluster interconnect we have the a single lif that is configured for the cluster management SVM. This lif is a lif that is always available in the cluster. If the node that hosts the lif fails, then that lif will failover to another node in the cluster. It is best practice to connect to that lif when managing the cluster.

Every node in the cluster has a node-management lif. Instead of connecting to the cluster-management lif, you could connect to the node-management lif to manage the cluster. However, the node-management lif will not failover, so Netapp advises not to manage your cluster via this lif.

The Service Processor of every node has its own ip address. The service processor can be used to manage the node remotely without losing connection when you for example upgrade a node, or troubleshoot a node. You can for example shutdown or reset a node from the service processor. Unfortunately we have no service processor in the simulator so we cannot have a lab on that.

And finally we ofcourse have datalifs to use in the cluster. These can be configured to failover to other nodes in the case of NAS lifs. SAN lifs do not failover.

Then we have ipspaces. By default, a multi-node cluster has 2 ipspaces: the Cluster ipspace and the Default ipspace. The cluster ipspace holds exactly 1 broadcastdomain the has the cluster interconnect ports. The default ipspace can have more then 1 broadcastdomain and you could have multiple data-ipspaces so to say. The advantage of having multiple ipspaces would be that you could have customers with identical subnets. By default, however, you only have the cluster and the default ipspace. So, if you have no need for duplicate subnets, your cluster can very well function with just the cluster and default ipspaces.

A broadcast domain is a collection of ports in an ipspace that are grouped together.

A very important thing to realize is that an SVM is always connected to an ipspace at creation time. This cannot be changed later on. So if you connect an SVM to an ipspace that it should not have been connected to at creation time, you will have to delete the svm and recreate it.

The cluster svm, is the svm that is setup during clusterconfiguration and this svm is the only one that is connected to the cluster ipspace. Do not confuse the cluster svm with the cluster-mgmt svm. I know this is confusing, but we will have a closer look at that in a second. (do not forget to show that in the lab!!!!)

==========

demo

So let's have a look at networking in our cluster.

net port show -node cl1-01

with net port show for node1 we see that we have port a upto and including port f. A and b are in the cluster ipspace and cluster broadcast domain. The rest of the ports are in the default ipspace and broadcast domain.

net int show -vserver Cluster -fields ipspace,address,home-port,home-node

when we run the net int show command for the Cluster svm with for the relevant fields that we want to see, we notice that we only use port a and b of every node and that these ports all belong to the cluster ipspace.

net int show -vserver cl1 -fields ipspace,address,home-port,home-node

when we run the same command for svm cl1, we notice all the management lifs and we see that they
live in ipspace default

net int show

running net int show without vserver or fields, we see all our lifs including the datalifs that belong to the datasvms.

net int show -fields ipspace

running vserver show -fields ipspace will tell us that the all svms are connected to the default ipspace
except the Cluster svm that hosts the cluster interconnects

vserver show -fields ipspace

And if we list the svms and to which ipspace they are connected we see that the only svm connected to the cluster ipspace is cluster.

==========

The last thing that we will have a look at in this overview is the different shells that you should be aware
of before we continue with module 2.

It is very simple: we have three different shells. The first one is the clustershell. This is the shell that you will be in during cluster administration, most of the time. Whatever you do in this shell, will
reflect in the cluster as a whole. So the RDB's will be updated whenever you create a volume or lif or user
or when you remove things. This shell can be accessed connecting to the cluster management lif or to any

of the node management lifs.

The second shell is the node shell. This shell will give you access to the node itself. So your commands
in that shell will not update the configuration of the cluster. For those of you that have worked with the
previous solution - 7-mode - you will be able to run 7-mode commands. Just keep in mind that what you do
in that shells effects the node only, and not the other nodes in the cluster. So from the clustershell you
can connect to the nodeshell of any of the nodes in the cluster.

And then there is the systemshell. This shell gives access to Unix. And also very important, this shell also
only effects the node that you access. To access the systemshell you will have to unlock the diag user and
give that user a password. Logging into the systemshell requires diag mode and you login as the diag user.

==========

Let's have a look at that.

We log into our cluster and from the clustershell we connect to the nodeshell of node1.
Now when we run vol show, we get an errormessage because this is not the clustershell. In the node shell
we should run vol status. The only volumes that we see are the volumes that are local to this node.
We do not see the other nodes vol0 and we only see one of the svm-root volumes. the other svm-root volume
is obviously in a different aggregate on a different node.

To exit this shell we type controld or exit.
To unlock the diag user we go to the diag mode and run security login unlock -username diag
then we give the user a password
and we can login to the systemshell by specifying the node we want to login to.

==========

Finally, there is one thing we definitely have to look at in this overview module. We have to be aware of how the data traverses the cluster.

The node in fact is made up of some software modules. These modules process the data that enters the node. The first modules are the network and SCSI modules. The network module is responsible for NAS traffic and the SCSI module is responsible for SAN traffic.
Then there's the Cluster Session Manager, this module is responsible for determining where the data should go. This means: is the data to be sent to the data module of this node, or does have to be sent to another node in the cluster. The data module is responsible for writing the data to disk. We could say that this is part of WAFL. So actually, the data module takes care of NVRAM and writing the data to the actual volume.

In the first scenario, the data enters the node and is destined to be written to a volume that is hosted in an aggregate on that particular node.

In the second scenario, the data enters the node and is destined to be written to a volume that is hosted in an aggregate on a different node in the cluster.

So the data enters the lif, let's say it is NAS data. The network module receives the data and copies it to the cluster session manager. The cluster session manager checks the RDB that deals with volumes: The volume location database. It sees that the volume is local to the node, so it copies the data to the data module. The data module stores the data in NVRAM, creates a stripe when it is time, adds parity and writes the data to the volume. After the data has been written, NVRAM can be flushed.

In the second scenario, the data enters the Network or SCSI module, is copied to the session manager and the session manager checks the VLDB. Now the volume is not local to the node so the data will be sent to another node via the cluster interconnect. The other node will copy the data to its data module and the NVRAM and the writing of the data is taken care of by the other node.