Architecture and Concepts

Hi and welcome to module one. This is the overview of module. First, let's have a look at the supporting 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, so in other words, you are not redundant to be redundant at the node level. You can add in to the existing cluster to form an ha pair and node in an ha pair shares the disks with the other node in the same ha pair. If one of the two notes fails, then the other node can take over the aggregates of the failed node. If you run an ha pair, you will have something which is called the HPA interconnect. This Han to connect is used to mirror the the ram between the two notes so that if one node fails, the data that was not yet written to disk is also available on the other node.

After having taken over the aggregates of the failed node, the data will still 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 volume data between the nodes of a cluster. It's a class that can be scaled out by adding more ha parents to the same cluster. Each aged care will have to be connected to the cluster interconnect and each ha pair will have its own HTA interconnect. Now 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 type to use. Currently with all flash arrays you can have a maximum of up to six ha pairs, which is 12 nodes. The configuration limits of all sorts can be found at the hardware universe age Dwu, netapp.com so we can have a single note with one or more details.

We can have a single age, a pair with one of modest shells or we can have multiple haps clown data and configuration data in an old type of environment is stored in volumes and volumes are always part of an aggregate and you could see, you could say that an aggregate is a collection of disks. In this picture you see for aggregates to road here gets 102 date aggregates. Also one per note, Agora zero note one is the root aggregate of note one and I go zero. No two is the root aggregate of no two and we see two data exits. Of course you can have multiple data aggregates in the cluster, but you can only have one root aggregate or one egg a zero per node. In the case of a node failure, the surviving node in the Ha pair will take over all of the aggregates of the other note by default.

You could prevent that when you hold the note manually if you wanted to. So you can specify a parameter that says that you do not want these disk groups or these aggregates to fail over. When the node joins the cluster, again, it can automatically or manually get it aggregates back. This is all configurable. There's a very important volume that resides in the road aggregate. There's volume is typically named Vole zero. This volume is the node volume and contains cluster configuration and log files, so every node has a volume named four zero. Another important feature of the Entreposto configuration is storage virtual machines. A cluster will always run storage. Virtual machines spms are logical representations of one of four types. We have the system types and there's only one of that type in our cluster. This SVM is used to manage the cluster interconnect interfaces. We have the note type and each node in your cluster is represented as a note as fame.

The Edmond type is the management Svm. You will usually connect to this storage virtual machine if you want to manage your cluster. So the Edmond type SBM basically represents your cluster up and finally we have the data type. Depending on the number of nodes and the protocols used, you can have up to a maximum of 1000 spms. This type of Svm is used to present data to our nest and sent clients data storage. Virtual machines are the spms that service the data. The data of the SPM is stored in volumes and all of the volumes that belong to a particular data SVM apart of the namespace of that particular SVM. So every storage virtual machine has its own name space. So in short, the namespace of an SBM contained all of its data volumes. A data SVM will always have one and exactly one root volume.

This is a very small volume of one gigabyte that is commonly used to mount the data volumes to these map points are called junction paths. In a nas environment, the volumes will have to be mounted, otherwise they will not be accessible by clients. This root volume of the storage virtual machine should never contain data. The only thing you will find in the storage virtual machines, root volume is or out of junction paths of the mounted the data volumes. Now let's have a look at cluster configuration. The configuration of your cluster is stored in something which we call the replicated databases. These replicated databases are in fact they're ASCII files, but they are in sync in the entire cluster. They're stored in vault zero and zero is the root volume of the node. If have four nodes in your cluster, you will have four vol Zeros.

So every node will have one and exactly one four zero. These databases should always be in sync in the classroom. So this means that all nodes will have the same information in their vault zero where it concerns the replicated databases and this concerns stuff like uh, volumes, Ip addresses, aggregates and aggregate information and what have you. So it's important to know as well that the root volume or vol zero is mounted to the slash en route directory on the node. You should not confuse vall zero with the road volumes of data storage virtual machines. Now before we have a look at the command line, let's have a look at what these RDB is really are. There are five of them and they all contain configuration information. The field to be contained, Meta information about volumes and aggregates become the or the block configuration and operations management. Daemon has set information. Think of I groups in such a fifth manager is for logical interfaces and networking as crs or the configuration replication servers is for replicating storage, virtual machine config to other clusters. And so think of Spm, Dr, things like that and management is for all the things that are not stored in one of the previous work so you could think of users, policies, et cetera.

And the location is listed here. We will definitely go and have a look at that later on in a, in another module.

Okay, so let's have a quick look at what we just discussed first. Um, I'm going to log into my cluster, which is via one 92 one 68 for 100 and I'm going to log in as Edmond. After I've logged in, I'm going to list my nodes. So I run close to show and it tells me that I've got four nodes in my cluster because their simulators, we can't say that we've got to ha pairs. Uh, unfortunately the simulators are in a gnome chair disc environment. So each node has its own disks in real life. This four node cluster would consist of two 80 pairs. Now let's check the storage virtual machines. We run veto of a show and we see that we've got seven Svms for notice vms, one admin SPM and two data svms. Now if we go to the advanced mode and run the same command again, we see yet another SPM. This is of the type of system. Um, so this is the SPM that is a responsible for the cluster interconnect interfaces.

Okay, so we have a look at these cluster interconnect and the other network interfaces. After we did the networking of a view to see all the volumes, we run volume show and we see that each node has its own vault zero. Also each vol zero has its own aggregates in which it lives. And the data volumes are in data elements. Obviously we have one, a node one Agur one, and we've got one new to Agora one. Both of those virtual machines have one root volume and one data volume page. So we exit and we will continue with networking.

Networking obviously is a very important item in the cluster configuration. First of all, if you run one or more ha pairs, there is this cluster interconnect. This can either be a switch if you've got one ha pair or it can be switched if you have more than one ha pair. You can also use a switch if you've got one ha pair, so that's up to you. Each node will have a minimum of two 10 gigabit interfaces and both interfaces will be um, uh, hosting an IP address and connected to the cluster interconnect network. These ports are configured during the cluster initialization or when a node joins the cluster. If needed, you can add more interfaces to the cluster interconnect network to grow the bandwidth. For example, next to the cluster interconnect, we have a single leaf that is configured for the cluster management SPM. This lift is live that's always available in the cluster.

If the node that hosts the lift fails, then that live will fail over to another node in the cluster. It's best practice to connect to that particular live when managing the cluster. In my example, just a second ago, I connected to 100 and that 100, um, IP address was my um, plus the management interface. Every node in the cluster has a node management live. Instead of connecting to the cluster management live, you could connect to the note management live and manage the cluster. You would end up in the same class, the shell and there would be no difference whatsoever. However, if the node management live, I would fail because the node fails that live would not fail over. So NetApp advises you not to manage your cluster and via the Node Management Lips, the server's processor of every node has its own IP address. The service persists or can be used to manage the node remotely without losing connection.

If you, for example, upgrade in node or troubleshoot in node so you can shut down or reset a node from the service processor and still have connection to your environment. Unfortunately, we have no service processor in the simulators, so we cannot have a lab on that. Finally, of course we've got the lifts to use in the cluster. These can be configured to fail over to other nodes. In the case of Nestle lips send lives will not fail over to other nodes. Then we've got IP spaces, which is relatively new and by default and multi-node cluster has to IP spaces, the cluster IP space and the default Ip space. This cluster IP space holds exactly one broadcast domain and um, that broadcast of May will host the cluster interconnect ports. The default IP space can have more than one broadcast domain and you could have multiple data IP spaces as well.

So you could have a number of ips basis and each IP space would have one or more broadcast than mates. The advantage of having multiple Ip spaces would be that you could have customers with identical subnets. So by default, however, you only have the cluster and the default IP space. So if you have no need for duplicate subnets, your cluster can very well function with just the cluster and default Ip spaces. I customer, and by the way is a collection of ports in an Ip space and these ports are grouped together in the broadcast customer. Very important thing to realize is that an SPM is always connected to an IP space and this is done at creation time. So when you create a story of

virtual machine, you can select the Ip space to which you want to connect to the Svm and you cannot change that later on.

So if you connect an Svm to an IP space that it should not have been connected to, then you will have to delete the storage virtual machine and recreated the cluster. SPM is the Svm that is set up during cluster configuration and this Svm is the only one that is connected to the cluster IP space. So the clusters frame holds the cluster interconnect ports and they are part of a different IP space. So obviously the cluster Ip space do not confuse the cluster SVM with the cluster management Svm. Okay. I know it's confusing, but we have a closer look at this in a second. So let's have a look at networking in our cluster. When we run net poured show for node one we see that we have poured a up to an including port F A and B are in the cluster IP space and cluster broadcast domain.

The rest of the ports are in the default IP space and broad customer. And then we run that in show for the cluster SPM and we want to list the Ip space, the address, the home port and home note. And then we noticed that we only use port a and B of every node and that these sports all belonged to the Gluster Ip space when we run the same command. Now for Spmc Oh one, we noticed all the management leaves and we see that they live in the Ip space default when we run net in show without visa or fields. Um, we see all our lives including the data lives that belonged to the data svms.

And if we run net in show and just want to see the Ip spaces, then and that will tell us at all his feelings are connected to the default IP space except the cluster Astrium that hosts the cluster interconnect. And if we list the spms to which Ip space they are connected, then we see that the only Ustream connected to the cluster IP space is the cluster SPM. The last thing that we will have a look at, uh, in this overview is the different shells that you should be aware of before you continue with module two. So let's have a look at the different channels that Netapp offers. It's very simple. We have three different shells. The first one is the cluster share. This is the share that you will be in during cluster administration. Most of the time, whatever you do in this shell will reflect in the entire cluster.

So the RDBMS will be updated when you create a volume or live or user or when you remove things or modifying things. This shell can be excess connecting to the cluster management live or to any of the node management lives. The second child is the node shell. This shell, we'll give you access to the note itself. So your commands in that chair will not update the configuration of the cluster. For those of you that I've worked with a previous solution of [inaudible], uh, which is seven mode, you will be able to run several mode commands. Just keep in mind that what you do in that chair effects the node only it doesn't affect the cluster. So from the classic chair, you can connect to a node shell of any of the nodes in the cluster. And then there's the system check. This shell gives you access to Unix and also very important, the shell only effects the node that you excess. So to access the system shell, you will have to unlock the diag user and give that user a password and logging into the system. Jail requires diag note and you have to log in as the direct user.

Okay, so let's have a look at that. We log into our cluster and from the cluster shell we connect to the node shell of node one. Now when we run those show, we get an error message because this is not the class this year that we're in. We're in the know Chet and in the Nacelle we should run vole status. The only volumes that we see are the volumes that are local to this node to node one. We do

not see any of the other volumes. We don't see evil zero of any of the other nodes. And we only see one of the Svm root volumes. The other Svm root volume is obviously in a different aggregate on a different note. So to exit this Shell, we type control D or exit. Now to unlock the Diet user, we go to the direct mood first.

Okay. And then we ran security login, unlock username Dayak, and then we give the user a password. And finally we can log into the system shell by specifying the node we want to look into. And if we want to return to the cluster shell, we type exit.

Okay, leave the shell and we're back in the glass shell. Finally, there's one thing we definitely have to look at in this overview module. Uh, we have to be aware of how the data for versus the cluster. When we take a closer look at the node in a cluster, we see that in fact it is made up of some software modules. They used to be called blades, but now they're called modules. But basically this is software. These modules processed the data that enters the node. The first modules are the network and scuzzy modules. Uh, obviously the network model module is responsible for NASA traffic and the squeezy module is responsible for send traffic. Then there's a class session manager and this is a very important module because it's responsible for determining where the data should go. This means is the data to be sent to the data model of this local node or does it 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 last module is part of WAFL, so actually the data module takes care of NPRM writes the data to the actual volume and then flushes and BRM. Okay. In the first scenario, the data interested in node and is destined to be written to a volume that is hosted in an aggregate on that particular note. 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. In the first scenario, the data enters the live, let's say it's Nesse data. The network module receives the data and copies it to the cluster session manager. The city manager checks the RDB that deals volumes, which is the VLDB or volume location database and that 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 MV Ram creates a strike when it's time to do that adds parity and writes data to the volume. After the data's been written and we ram is flushed. In the second scenario, the data enters the network or cause you module is copied to the session manager and the session manager checks the VLDB. Now the volume is not local to the notes, so the data will be sent to another node via the cluster interconnect. The other node will copy the data to state our module and the MV Ram, and the rest is taken care of by their data module on the other node. Now I hope this overview module was helpful, or at least not too boring, but if you are familiar with all of this already, but just to make sure that we're all at the same level, uh, before we start talking about booting your system and doing some troubleshooting. So hope to see you in the next module.