I'm Dr. Mann and I'd like to welcome you to
this Security+ Certification Exam Review
and we're going to take the
topic tonight - networking basics
and we're going to go ahead and start
with this first question on web servers.

A website that you have been trying to
serve has been taking a while to load and
s very laggy due to the large amount of HTTP
requests that are being sent to the web server.
Which of the following would be the best course
of action to improve the current and future
web server performance? Take a moment look at the
responses and then let's see what you're thinking.
So we got responses coming in the chat with a
lot of votes or Choice number B or Choice letter B.

So if we look at these answer choices adding
more RAM clearly does not help the situation.
The one that most people have chosen had
a second identical web server plus a load
balancer in front to spread out incoming HTTP
connections across. That is the correct answer.

So load balancers typically placed in the DMZ
generally used to load balance requests to web
applications but they can also be found inside
the network. Let’s see yeah which the phone
would be the best course of action to improve the
current and future performance so configuring SSL
on the web server to filter incoming is not
going to help us; configuring a router between
the web server and the default gateway not going
to help us either; so the correct answer is B.

Next question:

Which of the following is
true regarding IPv6 addresses?
And a lot of votes for D it
looks like all options are correct.
So if we look at these choices um IPv6
addresses are written using hex notation.
The first 64 bits contain the host's interface ID.
Maybe that could depend on which
way you are looking at the address
however generally going left to right
that will not be the host's
interface ID so let's see.

Take a look at this example address.

What you see highlighted in yellow is the site

prefix the subnet ID follows in the fourth
segment with quartet if you wish and then
the last 64 bits are the host interface ID.
So these are backwards and so all options
are not correct; the correct answer is C -
IPv6 addresses are written using X notation.

Let's see and the next one

You should use the command “ping 0:0:0:0:0:0:0:1”
on a Windows 10 server to determine what? What do you think?

Got responses in the chat; we've got votes
for A, C.
So this is pretty easy.
I think it's pretty clear that this is not
nor does it look anything like an ip4 address
so right away we can eliminate choices C and
D. That leaves A and B
and so basically we are looking to determine
if the local ipv stack is functioning.

So the question for you - how
else could you see this written?
so we have a response
Repeat the question.
Yes, how else could we see the command given in the question written?
This seems fairly cumbersome to write out
“ping 0:0:0:0:0:0:0:1” Etc.

Ping localhost?
Okay. How about
if we were going to use IPv6 address notation?
:: as in ping ::1. So remember that.
We can use leading zero suppression and
consecutive quartets or segments of all
zeros can be eliminated and replaced with the
double colon.
Let's go on to the next question.
Which IPv6 address is used for link local unicasts?
We've got some responses coming
in the chat and remember as you're
looking at answers or questions look
for things that you can easily eliminate.

And so far we have one response.
So we have one vote for Choice D; anybody else
wanting to weigh in before we
give out the answer on this one.
Is there anything that looks obviously incorrect?
A?
Yes. Very good so this
is the unspecified address

or the representation which indicates the absence
of an address and cannot be assigned to a host.
If you look at ::1
and think about from the last question
that's localhost and then FF00::/8 represents the IPv6 multicast.
The correct answer is Choice D - FE80::/10.This is link local.

Moving on.

What can ICMP be used for?
Let's see what's in the chat - we have lots of responses coming in and
looks like Choice D is because everyone's
selection here so all options are correct.
And choice D is the correct answer
so you can report IPV4 errors and Status IPv6
errors and status and this protocol supports the
function of troubleshooting utilities such as ping
and trace route these are all correct answers.
So what does ICMP stand for?
Internet Control Message Protocol.
Yes it does; very good
All right let's move on to the next question,
Identify all things wrong with this load
balancer diagram.
All right great so let's
take a moment look at this
and so our answer choices I'm going to read
these to you as we're looking at the diagram.

The first answer choice and by the way for this
question we are supposed to choose all that apply.
So Choice A load balancer is using a link local address
Choice B the left-hand web server IP is invalid
Choice C the middle web server IP is invalid; and
choice D the right hand web server IP is invalid.

So once again our four answer choices are
the load balancer is using a link local address
the web server to the left has an invalid IP
the one in the middle has an invalid IP and
the one on the right has an invalid IP

What do you think?

So we have a couple of answers in the in the
chat; let's get a couple more before we open it up.
So we have three responses A, A, B, D, C is incorrect.
All right so the load balancer
is using a link local address and that's
a problem; we would need a public address
00:13:55.260 --> 00:14:00.240
FE80 is linked local

What about the web server on the left here?

does anybody know what’s wrong with this IPv6 address?
This is the one that reads two zero
zero one colon colon c8o colon colon f82
so what's wrong with that is missing the last part of one part of the address
so you think it's missing something? Yes anybody else?
Well the this is definitely a problem and the reason is that you cannot use
the double colon replacement twice in the same IPv6 address
How about the web server in
the middle? does anybody know what's
wrong with this IPv6 address?
it has two wrong letters. That’s gone too far.

Gone too far - yes that is absolutely correct
and those letters are N and G so in
hexadecimal notation we use the characters
zero through nine and the letters A thru F.

Finally the server the web server
on the right? what's wrong with that address?

This may be the trickiest answer of the four
so I would direct your attention to the very last quartet.
How many characters?
I mean sure this last quartet or segment whichever you prefer
F8221- so there's one too many
hex characters there right there can only be four.
To me this is kind of like if you've seen those
I don't know what they are puzzles maybe where
you know they you're asked to read a statement
you know asked what's wrong with it and it's
one of those situations where they put the
word The in twice and unless you've seen it a lot
most people just miss it and read right over it;
I think maybe that's why the two twos
consecutive twos are there so kind of fools
your eye and too many characters in the
last segment yes so the correct answer here
is all four answers identify something
wrong with that load balancer diagram.
Let's go on to the next question:

Here we go:

You need to implement a load
balancer scheduling algorithm
to evenly spread active user sessions across
all web hosts. Which one should you choose?
So it looks like we've got a vote
for D and a few votes for A so
let me see A’s. yeah mostly so most people
are right now are selecting answer Choice A
and that is not correct.

Anybody else want to weigh
in before we go over this?
With a round robin algorithm it's absolutely
one of the simplest and as client requests come in
they're basically distributed one after the other
to each server and weighted round robin is
kind of the same thing except certain resources
have weights assigned to them and they're going
to be based on things like their processing
power and random yeah that's I would
say that's the the obviously incorrect answer. But
the correct answer here is the least connections
and in lease connections, load balancing takes into
account the current number of active connections
on each server the load balancer forwards the new
request to the server that is currently serving
up the lowest number of active connections. So that is the correct answer
for this question.

Let's go on to the next question -

Your IT manager says that your company's website
ordering system has a single point of failure in
the single load balancer then continues to suggest
configuring two load balancers where only one at a
time is in service. What type of load balancing
configuration has your it manager suggested?

Take a moment look at your
choices and make a selection.
We've got some responses
coming in; let's see what you're thinking.
So we have a lot of votes for Choice D
active passive and this is the correct answer.

Active-active is a configuration where both load
balancers are online; round robin and least
connections doesn't exactly apply to this; and
so active passive is what we're looking for and
with active passive one load balancer
handles the work until there's a problem.

And when there's a problem the standby load
balancer notices this and takes over the work.

Let's go on to the next question.

if your standby load balancer determines
that the primary load balancer is down
and you are operating on an active
passive load balancer configuration
what attribute would your standby
load balancer take control of?
So as you look at the answer choices

it's pretty clear that we're looking
for either a MAC address or an IP address.

So which do you think that would be?
IP address?
That is correct.

So we can eliminate the answer
choices A and B and that leaves choices C and D.
Let's see if we have any yeah we have a
couple of votes for C - load balancer IP address
and that is correct so in both active-active and active-passive
configurations
the load balancer has an IP address
assigned and we may know this or it's
also known as the virtual IP and so clients are
configured to send requests to the virtual IP
address assigned to the load balancer.
All right let's go on to the next question -

An I.T specialist for a
public-facing web application
wants to lessen the amount of processing
that each back-end server pool has
if they are using a back-end Windows server
configured in a load balancer and HTTPS is
already configured on each server what should
the IT specialist do let me take a moment
and look at your responses and then we'll
see what you're thinking on this question.

So we have some responses vote for C, one for B.
So Choice letter C
Choice C generate a new certificate is
not really going to do anything to help;
reduce the amount of processing on the
back end server pool so C is incorrect;
B
So the correct answer here is
B and that is to configure SSL TLS
termination at the load balancer. Choice A
says configure ipsec on the load balancer
ipsec operates at the network layer of
the OSI model which is responsible for
routing data packets and SSL operates at
the application layer of the OSI model.

Configuring SSL TLS termination at the load
balancer this should be the way to handle this
to handle SSL encryption decryption so the traffic
between the load balancer and the backend server
is then in H or represented as HTTP; SSL
should terminate at the load balancer
because it offers a centralized

location to correct SSL or Tax
and in order to perform deep packet inspection SSL
must be terminated at the load balancer or earlier
but traffic between the load balancer and
the application servers would be unencrypted.
And that's really what's going to get
you the Improvement in processing in
the amount of processing. The last choice
of using or enabling SSL TLS pass through
and so SSL passed through basically passes
data through a load balancer to the server
without decrypting it so that really doesn't
help us. So the correct answer here is B.

Next question-

Which of the following is not true
regarding OSI layers with load balancers?
So we have a response in the chat.
No idea
All right so let's take a little information about persistence and affinity.
Session persistence means that a user's requests
are consistently handled by the same server
or node during a session and session affinity
means that the requests are preferably handled by
the same server or node but not necessarily.

So based on that information
does that help you come to an answer choice?
Exactly
All right so which of the
following is not true then?
or are all answer choices true?
So all right let's see what we have in the chat
Here.
I'm going to explain one more time sure;
so again session persistence means that
a user's requests are consistently handled
by the same server or node during a session.

Affinity means that they are preferably handled
by the same server, but it's not necessarily true.
So based on this um first of all affinity
can be considered an OSI layer or approach
and persistence can be considered a
layer 7 approach so these two choices are true
the real question is here in Choice C the
statement affinity is more reliable than
persistence now remember persistence
is associated with the same node
affinity means that I would like that condition
to exist but it may not necessarily be true

So based on what we've said about persistence
and Affinity we have answer D
and another vote for d and
no the correct answer is C

Affinity is more reliable than persistence
this is the statement that is not true.

All right so while both have advantages typically persistence
is more reliable than session Affinity because
it ensures that all requests from a
client are handled by the same server.
So this is the best answer for this question.
There can be a little bit of confusion
sometimes between the terms persistence and
affinity - the names however are are
fairly good in that if something persists
it stays the same an affinity for something is
more like a desire so this this definitely
is an area of topic study that will help
you certainly with this exam but again I
think it's important that you remember persistence
typically means that requests are handled by the
same server affinity is that you would like that
situation to occur but it's not guaranteed.

Next question -

Which type of load balancing is
being depicted in the attached image?
I'd like you to take
a moment to look at your choices.
and then I'm going to bring up the image
so we can look at that which type of load
balancing that's what we are trying to determine.

So your answer choices are active active
least connections random or round robin
and I'm going to bring up the image.

Here's the image web clients client
requests enter the network load balancer.
You can see the requests are numbered one
two three and four and coming out of the
load balancer we see one and three going to the
top server and two and four going to the bottom.

Your choices for which type of load balancing
is being presented are active-active, least
connections, random, or round robin.
We've got selection of responses in the chat
and it looks pretty much like
everybody's saying A for active
active and one vote for least connections.

All right so client request one enters the load balancer goes to the top; two comes in goes to the bottom; three comes in goes to the top four comes in goes to the bottom. One, two, three, four.

Anybody want to uh change their opinion because active-active is not the correct answer.
So that leaves least connections, random and round Robin; Of those choices I would say that this does not look random to me - one two three four - that does not look random at all and right you've got several votes for round robin and that's exactly what it is.
All right so round robin is what was being depicted in that image.
Can I ask you a question?
Sure.
How is that scenario so much different from least connections?
Because the requests have been distributed evenly and in this sort of round robin fashion so the first one goes to the top server second one goes the bottom third to the top fourth to the bottom right so we're not looking at who's handling the least number of connections; we’re simply assigning the request top bottom top bottom top bottom right and you know it would be the same if you had more servers in fact if you had four you would see request one two three and four and if another request came in five it would start back up at the first server again and we would proceed in that manner round robin.

So basically iteratively one after the other and that's what they're depicting here all right.
So you know and again it from the load balancer’s point of view the load balancer knows about and has say a list for example of servers and then simply goes through them the list in order sending the requests down the list of servers and then when it gets to the bottom it just jumps to the top again and starts populating requests in order.

Let's go to the next question

Checking back-end server
responsiveness or availability -

Is the primary purpose of which load balancer option? take a moment look at your choices.
So I have a few responses coming into the chat and as always when you're looking at questions keep an eye out for something that may stick out as obviously incorrect

So you can eliminate it from the possible choices all right let's see what we have in chat.
So we have a few votes or A health probe,
all right so Source IP affinity really
doesn't fit and as we mentioned previously
um Affinity is you know were you trying
to route a request to the same target
but it's not guaranteed as opposed to persistence; round robin as we have seen is basically
traversing a list of targets in order and
fixed weight would be a weighted situation
but I don't really think any of those
have much to do with responsiveness
or availability and that actually leaves us
with the correct answer which is health probe.
And so the health probe is used to determine
the status of the instances in the back end pool
during load balancer creation you configure
a health probe for the load balancer to use
the help probe will determine if an
instance is healthy and can receive traffic
so A is the correct answer to this question.

All right, let’s move on to the next question.

While using Source IP affinity in a load
balancer configuration when a client establishes
a connection its session becomes blank to
the node that first accepted the request.
All right what do you think?

Yes, good.

We have a couple of selections; I think this one's kind of obvious

Let's see what you're thinking here
uh votes for B as the correct answer
and yeah that is correct. Source IP affinity
and lost does not fit the statement.
Detached does not fit the statement - those
two I think are the most obvious to eliminate.
Yeah really doesn't make sense
and knowing what we know about
Source IP affinity bound is the best
answer to this question;
Let’s move on.

One of your back end servers named WWW2 is used by a load balancer as one web head in a load balanced pool.
WWW2 has more RAM and CPU processing power than
others in the same web poo. To ensure it hosts
more client sessions than the others what
is the best load balancing algorithm to use?
Take a moment look at the

choices and make a selection.
Let's see what's in the chat
so we have lots of votes
for B which is least response
and this is correct. The key here is that the
Web head or the WWW2
has more RAM and processing power
than others so we know that we
wanted to handle more work because it's
going to be more capable of doing this.
So this question hinges on the difference
between least response and least connections.
So starting with Choice C random that's the
I would say the obviously incorrect answer;
and understanding that round robin simply
just follows a list iteratively in order makes
that unsuitable so it really boils down
to least response released connections
and the difference is that at least
connections moves - forwards - new requests
to the server that's currently servicing
the lowest number of active connections.

Least response takes into account active
connections plus the average response time
so that gives us that little extra edge
to to determine which you know algorithm
at least in this question is the best and
certainly in the reality of the situation
if you were using that then it's
going to choose the more powerful
load balancer. Let me see back when we
serves low balancer and web heads right
so we're looking for lowest number of
active connections and shortest response time
at shortest shortest average response time
so the correct answer is B least response.

Now we jump way back up here
Which of the following are not an advantage
from utilizing a load balancer? Choose two.

So we got a few responses in the chat uh
yeah votes for B and C so better app security
and better at regulatory compliance - these are the correct answers.
Really not advantages from utilizing load
balancing performance and better availability
so the two that are not are app
security and regulatory compliance.
and the next question and this is the last
one for this section…

After reviewing the feedback from your
customer satisfaction surveys you notice that

the predominant complaints is how saved items in a
customer's shopping cart are constantly being lost.
What load balancer configuration is
a solution to satisfy this complaint.
All right let's see what's in
the chat.

Let me see the three five lots
of votes or a session persistence
and this is the uh correct answer clustering and
loop prevention really don't have anything to do
with this and that leaves the two choices as a
recession persistence and database synchronization.
They are different concepts , session
persistence as we've discussed refers to the
ability of a system to maintain a user's
session data across multiple requests
session data can be stored in memory files
databases or other storage systems a database
synchronization refers to the process of keeping
two or more databases in sync with each other.
So the best answer is session persistence and
that ends this topic on networking basics.