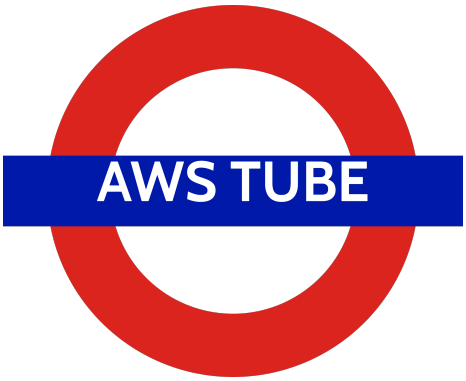

Figure 1: CLICK HERE TO ENTER STATION

Recovery Sheet 9

SJ Green

Recovering from the session delivered on the night of **Wednesday 19th March 2025**. You can view the recovery sheet for the previous week here.

1 Active-Active

Description

1. **Term 1**. Description goes here.
2. **Term 2**. Description goes here.
3. **Term 3**. Description goes here.
4. **Term 4**. Description goes here.
5. **Term 5**. Description goes here.
6. **Term 6**. Description goes here.
7. **Term 7**. Description goes here.
8. **Term 8**. Description goes here.
9. **Term 9**. Description goes here.
10. **Term 10**. Description goes here.

2 Warm Standby

1. **Overlapping CIDR block**. Two CIDR blocks are said to overlap if they share any common IP addresses.

2. **AWS Site-to-Site VPN**. AWS introduced the "VPN Classic" label on October 3, 2017, as part of an update to their VPN service. This update included features such as support for custom Pre-Shared Keys (PSKs) and inside tunnel IP CIDR blocks. Additionally, the AWS SDK was updated to include a new 'category' field, allowing users to identify their VPN connection as either 'AWS Classic VPN' or 'AWS VPN'. This differentiation provided insight into the capabilities of the VPN connection, with the new features being available only for 'AWS VPN'.

3. **AWS Direct Connect** Description goes here.

4. **VPC Peering**. Description goes here.

5. **Gateway VPC endpoint** Description goes here. "When your instances access Amazon S3 or DynamoDB through a gateway endpoint, they access the service using its public endpoint." states the User Guide. This is rather confusing, because it suggests that there is a public connection. If we are using public endpoints, we must be using a public connection, right? Yet the whole idea of VPC endpoints, I thought, was that they avoid the public Internet.

   Yes, your instances connect to the public DNS names of S3 or DynamoDB, such as s3.amazonaws.com or dynamodb.eu-west-2.amazonaws.com. But thanks to the Gateway Endpoint and some internal AWS magic, that traffic does not traverse the Internet. It stays entirely within the AWS network (i.e., private connectivity over the AWS backbone).

   Think of it as:

   "You are using a public DNS name, but AWS transparently reroutes your request privately via the VPC endpoint."

6. **Interface VPC endpoint**.

7. **AWS PrivateLink**. Description goes here.

8. **AWS Transit Gateway**. Description goes here.

9. **AWS Lattice** Description goes here.

10. **Term 9** Description goes here.

11. **Term 10**. Description goes here.

3 Pilot Light

The pilot comes aboard ships in unfamiliar waters to sort out shit.

1. **Overlapping CIDR block**. I think we should take a moment to clarify what people mean when they talk about two CIDR ranges *overlapping*. Rugs, napkins, sounds, and events can overlap. But in what sense can a CIDR range, such as 192.0.2.0/24 overlap with some second CIDR range? What do we look for to detect overlapping? Would that second CIDR range be one with an initial number which is smaller than 192, so it somehow "fits inside" it? For example 10.0.2.0/24. Or is it really about that number after the forward slash? Perhaps that second CIDR range will be overlapped if the number at the end, after the slash, is *smaller* than 24. Or must it be bigger than 24? So, you see, we need to clarify what we mean when we talk about two CIDR ranges which are overlapping.

   When we say that two CIDR ranges overlap, we mean that the two ranges share some portion of IP addresses. In other words, at least one IP address exists in both ranges. The question raised, then, is how we *know* that there is some IP address which exists in both ranges.

   The procedure is simple. First, we convert the CIDR blocks which we are comparing (e.g. 192.168.1.0/24 and 192.168.1.128/25) into *ranges*. I will use the first block as an example: 192.168.1.0/24 would become 192.168.1.0/24 - 192.168.1.255. I have turned that CIDR block into a range.

   Do you see how I did that? I considered **two things**. I first considered the fact that the first 24 bit positions are for the purpose of network identification. The "24" after the slash tells me this. I call this menacing thing **The Structural Slash**, since it tells me nothing about content - it does not mean there are 24 networks, it does not tell me anything about actual IP addresses - it only gives me structural information, telling me that the first 24 parking spaces are reserved. So, we have the final 8 bits to identify hosts. That's the final byte. I second consider the **content** of the final eight bit positions (the parking spaces). Sure, I have eight parking spaces to work with but is there anything *in* them? Well, it turns out that there is a 0. So, all eight bit positions are empty. So, we can use all eight bit positions to generate differing numbers. Each position can take two values. So, it's 2 to the power of eight. This gives me 255. I can create IP addresses for 255 hosts. Therefore, the range is 192.168.1.0 - 192.168.1.255.

   Smaller numbers after that Structural Slash at the end command larger numbers of IP addresses. This is quite counterintuitive, so I think it is worth elucidating. A CIDR block denoted with /8 has a *larger*—yes, larger—number of potential IP addresses than a CIDR block with /24.

This is because the Structural Slash or Staff Slash denotes the bit positions to be used for network identification. The networks are the essential condition for hosts, so it's like a parking attendant laying out a sign–outside a hospital–declaring that 10 spaces are to be reserved for the doctors, since they are the staff and without doctors, no patients can be hosted. The staff slash denotes the number of parking spaces for *networks*, not for hosts. This makes sense; networks are important and akin to doctors. Add to this the seesaw principle, which means that if more staff are using up parking spaces, fewer patients are able to park and vice versa ("one goes up, the other comes down") and it's easy to see why *large* numbers for the Staff Slash, or Structural Slash, produce *small* numbers of IP addresses.

Suppose two napkins are laid out on a table for a dinner party. One way they might be overlapping is if a small red napkin is placed on top of a large white napkin. The red fits entirely within the white, perhaps to look like the flag of Japan. In this case, one napkin is a subset of the other. In most examples involving CIDR blocks, one is a subset of the other. All the IP addresses of one CIDR block are *covered*, we might say, by a second, larger CIDR block.

But perhaps it is a dinner party for Nepalese guests, and we are laying out two red triangular napkins, slightly overlapping one another, to look like the flag of Nepal. Both triangles are the same size. It is not the case that one is a subset of the other. It is just that a small area of the table is covered by both napkins. This situation is: overlapping, without one being a subset of the other. This can also occur with two CIDR blocks.

Let's look at an example. Consider the CIDR block 192.168.1.0/26. To turn this into a range, I calculate how many bit positions can be used for host IDs. 32 - 26 is 6. So, there are 2^6 possible IP addresses (64 addresses). Therefore, the range is 192.168.1.0 - 192.168.1.63. Now, you might be confused, thinking *surely it should be 192.168.1.**64***. It is not because we use zero-based counting. The first address is 192.168.1.0, the second 192.168.1.1... the 64th is 192.168.1.63.

The second block in our example will be 192.168.1.32/26. Turned into a range, this is 192.168.1.32 - 192.168.1.95. We do 32 - 26, which equals 6. So, 2^6 addresses again, or 64 addresses. But this time, the first address is 192.168.1.32 and the second 192.168.1.33. We can do 32 + 63 to give 95. So, the last address is 192.168.1.95.

So, consider whether these two ranges overlap:

`RANGE 1: 192.168.1.0 - 192.168.1.63`

```
RANGE 2: 192.168.1.32 - 192.168.1.95
```

They *do* overlap, Nepalese style. There is *partial* overlap, we might say. Consider the range 192.168.1.32 - 192.168.1.63. This is the range of concurrent coverage - the overlapping napkins. An address like 192.168.1.5 is in the first range only. An address like 192.168.1.90 is in the second range only. An address like 192.168.1.32 is in both ranges.

Now let's look at some Japanese-style overlap. *Think of the Japanese flag where the red circle is fully contained within a white background.* This is a case where one CIDR block denotes a set of IP addresses and they are a **subset** of the set denoted by a second CIDR block. We might call the smaller network, which is a subset, a sub-network, or *subnet* for short. Our first block will be 192.168.1.0/24. Our second CIDR block will be 192.168.1.128/25. The first block is the range 192.168.1.0 - 192.168.1.255. The second block is the range 192.168.1.128 - 192.168.1.255. Here, 192.168..1.128/25 is a subset of 192.168.1.0/24. Every IP address in the former block is also in the latter block.

Are there any rules of thumb when it comes to identifying overlapping CIDR blocks? This is where we need to be careful. Some people think that if two CIDR blocks have the same prefix length, then they necessarily overlap. But this is not true. Consider these two blocks:

```
192.168.1.0/24

192.168.2.0/24
```

The two blocks are both /24. However, they do not overlap. The first range is 192.168.1.0 - 192.168.1.255. The second range is 192.168.2.0 - 192.168.2.255.

Two CIDR blocks with the same number after that Structural Slash might overlap, though. Take these two blocks:

```
192.168.1.0/24

192.168.1.168/24
```

These blocks **do** overlap. The clue is that the prefix length is the same (both are /24) *and the initial address is the same.* That is, they both begin with 192.168.1.

If you ever come across a case in which the two CIDR blocks have the

exact same initial address, but a differing prefix length, then it will be a case of overlapping CIDR blocks. Consider these two CIDR blocks, for example:

```
192.168.1.0/24

192.168.1.0/26
```

Remember, again, that when that Staff Slash is large, there are many staff and few patients. There are a *few* IP addresses with the /26 CIDR block and *many* IP addresses with the /24 block. Here, the smaller /26 block is contained within the larger /24 block.

It's worth emphasizing that **just because two CIDR blocks have some numbers in common at the beginning, it is not necessarily the case that they overlap**. You see, the thought might be that with the two CIDR blocks 192.168.1.0/30 and 192.168.2.0/24 there is overlap. After all, they both begin with 192.168. "Because their names have a common element, the sets of IP addresses which they denote must have a common element" runs the thought. The conclusion is that these two CIDR blocks must have some IP addresses in common because they both begin 192.168.

I think we should take this misconception seriously. It is a very reasonable to suppose that if two blocks both begin with X then they denote similar things, or have something in common. Why else would would part of the name be the same if they did not denote the same sorts of things? George Bush and George Bush Jr have the same blood type. Homo erectus and homo sapien are similar. Surely two things both beginning 192.168 are similar.

Well, a CIDR block conveys not only a prefix length (the number after the slash) but a **starting address**. That starting address is usually called the network address. "Because their names have a common element, the sets of IP addresses which they denote must have a common element". What this person is doing, in this line of argument, is confusing the two names for names of IP addresses. Yet they are names of *blocks*. It is true that 192.168.1.0 is similar to 192.168.2.0. From the common element in these names, we can infer that both these IP addresses are part of the same network.

However, we cannot argue, in a seemingly similar vein, that the **block** 192.168.1.0 is similar to the **block** 192.168.2.0. When we argue like this, we mean that the blocks have some IP address in common. **Blocks** are containers for IP addresses, unsettled upon individual addresses. If you

consider an IP address name, the content of the name *does* often betray its homely network, the thing of which it is part. The IP address name 192.168.2.1 betrays its home, namely the network 192.168.2.0. So, the person mistakenly applies the logic of IP address names to the logic of CIDR block names. They think that the block name betrays something about the thing about which *it* is part. They continue: "if two blocks begin the same, they are part of the same thing." After all, if two IP address names begin the same, they are part of the same block.

Well, maybe. But the crucial response is this: Maybe they *are* part of the same thing, but that doesn't mean that they have any internal parts in common. CIDR blocks are isolated chambers. "Human stomach" and "human lungs" both begin with "human", and indeed both denote things *in a human*. But what is in the stomach is not in the lungs, and vice versa. What these two names denote does not overlap.

2. **AWS Site-to-Site VPN**.
3. **AWS Direct Connect** Description goes here.
4. **VPC Peering**. Description goes here.
5. **Gateway VPC endpoint** Description goes here.
6. **Interface VPC endpoint**. Description goes here.
7. **AWS PrivateLink**. Description goes here.
8. **AWS Transit Gateway**. Description goes here.
9. **AWS Lattice** Description goes here.
10. **Term 9** Description goes here.
11. **Term 10**. Description goes here.

4 Backup

1. **Overlapping CIDR Block**. Description goes here.

2. **AWS Site-to-Site VPN**.

3. **AWS Direct Connect**.

Matthews, Nick (2017). Connecting Many VPCs. *Reinvent 2017*. Available at: `https://www.youtube.com/watch?v=KGKrVO9xlqI`

Nick Matthews Presentation - Quiz 1. April 1st 2025. Available at: https://concentric-2903432.s3.eu-west-2.amazonaws.com/Firmament/2011/Direct_Connect/Nick_Matthews_Quiz_1.pdf

Nick Matthews Presentation - Quiz 2. April 1st 2025. Available at: https://concentric-2903432.s3.eu-west-2.amazonaws.com/Firmament/2011/Direct_Connect/Nick_Matthews_Quiz_2.pdf

Lehwess, Matt (2015). Double Redundancy. *Reinvent 2015* [Conference]. Available at: https://www.youtube.com/watch?v=_JgNnmOfxLE

Davies, Justin (2018). AWS Direct Connect Deep Dive. *Reinvent 2018* [Conference]. Available at: https://www.youtube.com/watch?v=DXFooR95BYc&t=37s

Hoekelman, Brian and Brooke Mouland and Roger Greene (2014). AWS Direct Connect Solutions and Network Automation. *Reinvent 2014* [Conference]. Available at: https://www.youtube.com/watch?v=m6NyVJrXUjo&t=452s

Seymour, Steve (2017). Deep Dive on AWS Direct Connect and VPNs. *Reinvent 2017* [Conference]. Available at: https://www.youtube.com/watch?v=eNxPhHTN8gY&t=91s

Green, Sam (2025). Confusions in the AWS Direct Connect Terminology. April 3rd 2025. *Medium*. Available at: https://medium.com/@samjackgreen/confusions-in-the-direct-connect-terminology-72d179d24713

4. **VPC Peering**

Barr, Jeff (2014). New VPC Peering for the Amazon Virtual Private Cloud. March 26th 2014. *AWS News Blog*. Available at: https://aws.amazon.com/blogs/aws/new-vpc-peering-for-the-amazon-virtual-private-cloud/

VPC Connectivity Options [Whitepaper]. Available at: https://d1.awsstatic.com/whitepapers/aws-amazon-vpc-connectivity-options.pdf

Adamski, Tom (2018). AWS VPN Solutions. *Reinvent 2018* [Conference]. Available at: https://www.youtube.com/watch?v=qmKkbuS9gRs

5. **Gateway VPC Endpoint**.

Morris, Gina (2018). Your Virtual Data Center. Reinvent 2018 [Conference]. Available at: https://www.youtube.com/watch?v=jZAvKgqlrjY&t=71s

6. **Interface VPC Endpoint** Description goes here.

   Weiss, Becky (2021). Securing your data perimeter with VPC endpoints. Dec 17th 2021. Available at: https://www.youtube.com/watch?v=iu0-o6hiPpI

7. **AWS PrivateLink**.

   Devine, James (2018). Best Practices for AWS PrivateLink. *Reinvent 2018* [Conference]. Available at: https://www.youtube.com/watch?v=85DbVGLXw3Y

   Morris, Gina (2018). AWS PrivateLink Fundamentals. *AWS New York Summit 2018* [Conference]. Available at: https://www.youtube.com/watch?v=20RxEzAXG9o

   MacCárthaigh, Colm (2017). AWS PrivateLink Deep Dive. Reinvent 2017 [Conference]. Available at: https://www.youtube.com/watch?v=8WhLO4y5-J0&t=3s

   MacCárthaigh Presentation Quiz 1. April 1st 2025. Available at: https://concentric-2903432.s3.eu-west-2.amazonaws.com/2017/PrivateLink/colm_maccarthaigh_presentation_worksheet_1.pdf

   Epshteyn, Ilya (2018). How Vanguard and Bloomberg Use AWS PrivateLink. Reinvent 2018 [Conference]. Available at: https://www.youtube.com/watch?v=63NG-s-2HdQ

8. **AWS Transit Gateway** Seymour, Steve (2018). Introducing AWS Transit Gateway. *Reinvent 2018* [Conference]. Available at: https://www.youtube.com/watch?v=yQGxPEGt_-w

   Matthews, Nick (2018). Transit Gateway and Transit VPCs. *Reinvent 2018* [Conference]. Available at: https://www.youtube.com/watch?v=ar6sLmJ45xs

   Advanced VPC Design and New Capabilities for Amazon VPC. *Reinvent*

2018 [Conference]. Available at: https://www.youtube.com/watch?v=fnxXNZdf6ew

9. **AWS Lattice** Description goes here.

   Introducing Amazon VPC Lattice. Dec 8th 2022. *Reinvent* [Conference]. Available at: https://www.youtube.com/watch?v=fRjD1JI0H5w

   Davies, Justin (2023). Amazon VPC Lattice architecture patterns and best practices. Available at: https://www.youtube.com/watch?v=zQk9AIPVdXs

10. **Term 10** Description goes here.