

Subnetting from CIDR Notation Project + Cheat Sheet (Worksheet Style)

Author: Oluwaseun Osunsola

Environment: AWS

Project Link: [GitHub Repository](#)

Introduction

This project demonstrates the process of subnetting an IP address using CIDR (Classless Inter-Domain Routing) notation. Subnetting is a fundamental networking concept that allows a larger IP network to be divided into smaller, more manageable sub-networks. This is essential for efficient IP address allocation, improved network performance, and enhanced security.

The given problem involves the IP address **192.168.60.55/20**. By applying a structured step-by-step method, we will determine:

1. The **Subnet Mask** corresponding to the CIDR value
2. The **Network ID** (the first address in the subnet)
3. The **Broadcast ID** (the last address in the subnet)
4. The **Usable IP Address Range** available for hosts within the subnet

This approach uses binary conversion, logical AND operations, and the concept of the "magic number" to simplify calculations and make the subnetting process easy to follow.

Step 1: Write Network Information

Before doing anything, write down what is given.

- **IP Address:** 192.168.60.55
 - **CIDR Notation:** /20 (this tells us how many bits are ON in the subnet mask)
-

Step 2: Subnet Mask

CIDR /20 means the **first 20 bits are 1s** (ON).

- Binary: 11111111.11111111.11110000.00000000
- Decimal: 255.255.240.0

🔗 This shows that the subnetting occurs in the **3rd octet** (since the mask breaks inside it).

Step 3: Convert Target Octet to Binary

Take the octet where subnetting happens (**3rd octet = 60**) and convert it into binary.

Bits Table:

```
128 64 32 16 8 4 2 1
```

Conversion by subtraction:

- $60 - 128 = \text{not possible} \rightarrow 0$
- $60 - 64 = \text{not possible} \rightarrow 0$
- $60 - 32 = 28 \rightarrow 1$
- $28 - 16 = 12 \rightarrow 1$
- $12 - 8 = 4 \rightarrow 1$
- $4 - 4 = 0 \rightarrow 1$
- Remaining $\rightarrow 00$

Binary of 60 = 00111100

🔗 This method shows exactly which bits are ON for the given octet.

Step 4: Logical AND with Subnet Mask

Now, compare the **IP octet** with the **subnet mask octet** using AND logic.

```
IP (3rd Octet):    00111100
Mask (3rd Octet):  11110000
Result:            00110000
```

🔗 This step zeroes out the host bits, leaving only the **network bits**.

Step 5: Convert Result Back to Decimal

Take the result (00110000) and convert back:

- $32 + 16 = 48$

🔗 The **network portion** of the 3rd octet is **48**.

Step 6: Network ID

Replace all host bits with 0s.

- **Network ID = 192.168.48.0**
-

Step 7: Broadcast ID

To find the broadcast:

1. Calculate the **magic number** = $256 - \text{subnet mask value in target octet} = 256 - 240 = 16$

2. Next network = $48 + 16 = 64$
3. Broadcast = Next Network - 1 = **63**

- **Broadcast ID** = **192.168.63.255**

🔑 The broadcast is always the last address in the subnet.

Step 8: Usable IP Range

The usable IPs are all addresses **between Network ID and Broadcast ID**, excluding both.

- **First Usable** = **192.168.48.1**
- **Last Usable** = **192.168.63.254**

🔑 These are the addresses you can assign to hosts.

☑ Final Answer

- **Network ID:** **192.168.48.0**
- **Broadcast ID:** **192.168.63.255**
- **Usable IPs:** **192.168.48.1 → 192.168.63.254**

Perfect ☑ — here's a **Conclusion** that ties your subnetting project together:

📌 Conclusion

In this project, we subnetted the given IP address **192.168.60.55/20** to determine its network parameters. By carefully applying binary conversion, logical AND operations with the subnet mask, and calculating the magic number, we derived the following results:

- **Network ID:** **192.168.48.0**
- **Broadcast ID:** **192.168.63.255**
- **Usable Host Range:** **192.168.48.1 – 192.168.63.254**

This shows that the **/20** subnet provides **4,094 usable IP addresses**, making it suitable for medium-to-large networks.

The method used ensures accuracy and consistency by breaking down subnetting into clear, logical steps. With practice, this approach can be applied to any CIDR-based subnetting problem, making it an invaluable skill for networking and cybersecurity tasks.

📄 Subnetting from CIDR Notation – Cheat Sheet (Worksheet Style)

Step 1: Network Information

Item	Value
IP Address	...
CIDR Notation	/__

Step 2: Subnet Mask

Form	Value
Binary	_____ . _____ . _____ . _____
Decimal	__ . __ . __ . __

🔗 CIDR tells you how many bits are ON (network bits).

Step 3: Convert Target Octet to Binary

Focus on the octet where the subnet mask breaks.

Bits Table:

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Binary Conversion of Octet ():

__	-	128	=	?	→	(0/1)
__	-	64	=	?	→	(0/1)
__	-	32	=	?	→	(0/1)
__	-	16	=	?	→	(0/1)
__	-	8	=	?	→	(0/1)
__	-	4	=	?	→	(0/1)
__	-	2	=	?	→	(0/1)
__	-	1	=	?	→	(0/1)

Resulting Binary = _____

Step 4: Logical AND with Subnet Mask

IP Octet:	_____
Mask Octet:	_____
Result:	_____

🔗 This gives the **network portion**.

Step 5: Convert Result Back to Decimal

Result (binary): _____ → Decimal = ____

Step 6: Network ID

____ . ____ . ____ . 0

🔗 First address in the subnet (all host bits = 0).

Step 7: Broadcast ID

- Magic number = (256 – subnet mask value in target octet) = ____
- Next network = Network Octet + Magic Number = ____
- Broadcast Octet = Next Network – 1 = ____

Broadcast ID = ____ . ____ . ____ . 255

Step 8: Usable IP Range

Item	Value
First Usable IP	____ . ____ . ____ . 1
Last Usable IP	____ . ____ . ____ . 254

🔗 Range excludes Network ID and Broadcast ID.

☑ **Final Answer**

- **Network ID:** ...
- **Broadcast ID:** ...
- **Usable IP Range:** ... → ...