

# ネットワーク基礎 - 1問1答ガイド

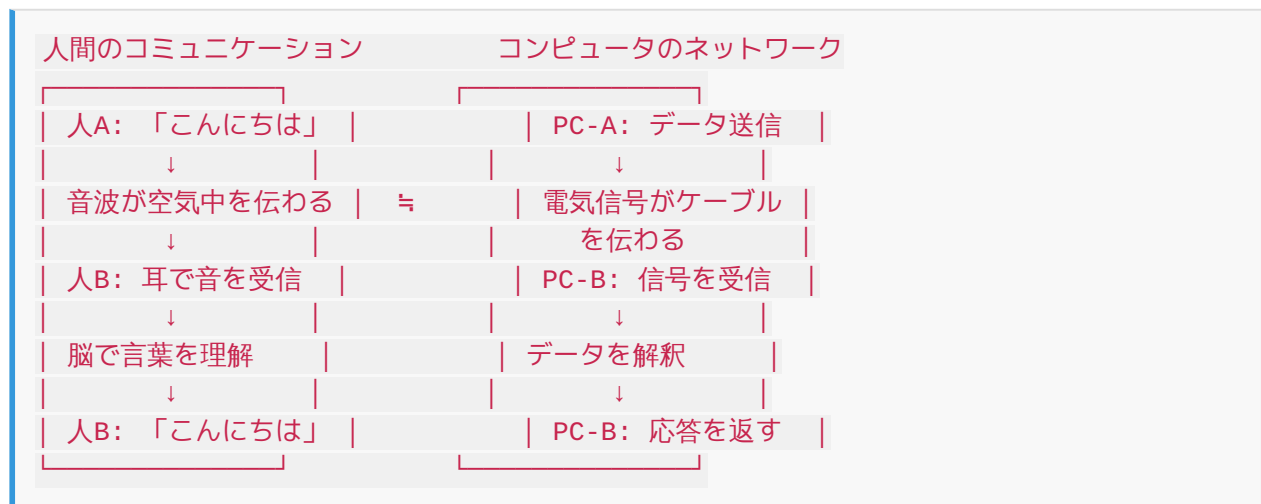
全く何も知らない人のためのネットワーク入門資料

## Q1: ネットワークとは何ですか？

**A:** ネットワークとは、複数のコンピュータ（PC）同士が接続して、情報をやり取りできる仕組みのことです。

人と人のコミュニケーションに例えると

ネットワークは、人と人が言葉で会話するのによく似ています：



**重要なポイント:** - 人間は「日本語」「英語」などの共通言語が必要 - コンピュータも「共通のルール（プロトコル）」が必要 - お互いが理解できる方法で通信しないと会話が成立しない

## Q2: 電気信号とパケットって何ですか？

**A:** コンピュータは、電気信号を使って通信します。この電気信号を意味のある単位にまとめたものがパケットです。

## 概念的な説明

### 【電気信号の流れ】



## パケットとは？

手紙に例えると分かりやすいです：

### 【手紙】

### 【パケット】

宛先：東京都...	宛先IPアドレス：
差出人：大阪府...	192.168.1.10
_____	送信元IPアドレス：
こんにちは。	192.168.1.5
お元気ですか？	_____
(本文)	データ本体

パケットの構成要素: 1. ヘッダー (宛先情報) : どこに届けるか 2. データ本体: 実際の情報 3. 送信元情報: どこから来たか

## 通信の流れ

ステップ1: PC-Aがデータを作成

↓

ステップ2: データをパケットに変換 (宛先などの情報を付加)

↓

ステップ3: パケットを電気信号に変換

↓

ステップ4: ケーブルを通じて電気信号を送信

↓

ステップ5: PC-Bが電気信号を受信

↓

ステップ6: 電気信号をパケットとして解釈

↓

ステップ7: パケットから元のデータを取り出す

↓

ステップ8: PC-Bが必要に応じて応答パケットを返す

つまり：ネットワークで「繋がる」とは、 1. 電気信号が物理的に届くこと 2. その信号をパケットとして正しく解釈できること 3. パケットに対して適切に応答できること

この3つが全て成立することです。

## Q3: ネットワークの歴史を簡単に教えてください

**A:** ネットワークは、軍事研究から始まり、大学や研究機関に広がり、最終的に一般家庭まで普及しました。

### 歴史の流れ

#### 【1960年代】ARPANET（アーパネット）

米国国防総省の研究プロジェクト



目的：核攻撃でも生き残る分散型ネットワーク

#### 【1980年代】TCP/IPプロトコルの標準化

→ 異なるコンピュータ同士が通信できる共通ルールができた

#### 【1990年代】インターネットの商用化

→ 一般家庭でもネットワークが使えるようになった

#### 【2000年代以降】

- 無線LAN（Wi-Fi）の普及
- スマートフォンの登場
- クラウドサービスの発展
- IoT（モノのインターネット）

**重要な進化のポイント：** - 最初は数台のコンピュータを繋ぐだけ - → 共通ルール（プロトコル）の確立 - → 世界中のコンピュータが繋がるインターネットへ

## Q4: IPアドレスとは何ですか？

**A:** IPアドレスとは、ネットワーク上のコンピュータを識別するための住所のようなものです。

前のQ2で学んだ「パケットには宛先が必要」という話を覚えていますか？その宛先を指定するのがIPアドレスです。

### 住所との比較

【現実世界の住所】	【ネットワークのIPアドレス】
東京都渋谷区...	192.168.1.10
↓	↓
この住所に手紙を配達	このIPアドレスにパケットを配送

### IPアドレスの形式

IPv4アドレスの例: 192.168.1.10

↓ ↓ ↓ ↓  
└──┴──┴──┴──┘ → 4つの数字（0-255）をドットで区切る

ネットワーク内のコンピュータ:

PC-1: 192.168.1.10	← 各PCに固有のIPアドレス
PC-2: 192.168.1.11	
PC-3: 192.168.1.12	
スマホ: 192.168.1.20	

## IPアドレスの種類

### 【プライベートIPアドレス】

家庭内・社内ネットワークで使用

例: 192.168.1.x, 10.0.0.x

自宅ネットワーク

192.168.1.10

192.168.1.11

### 【グローバルIPアドレス】

インターネット上で使用（世界で一意）

例: 203.0.113.5

インターネット全体で重複しない

**覚えておくこと:** - IPアドレス = ネットワーク上の住所 - 同じネットワーク内で重複してはいけない - パケットの送り先・送り元を指定するために必要

## Q5: ルーターとは何ですか？どんな役割がありますか？

**A:** ルーターとは、異なるネットワーク同士を繋ぐ「交通整理係」です。

Q4で学んだIPアドレスを使って、パケットを正しい宛先に届ける役割を持っています。

### 郵便局との比較

#### 【郵便配達の仕組み】

A市の郵便局

↓ 手紙を転送

中央郵便局（分配センター）

↓ 宛先に応じて転送

B市の郵便局

↓

受取人

#### 【ネットワークのルーター】

ルーターA

↓ パケットを転送

↔ インターネット

↓ 宛先IPに応じて転送

ルーターB

↓

宛先PC

## ルーターの具体的な動き

【シナリオ】PC-AからPC-Bにデータを送る



ステップ1: PC-Aが「10.0.0.50にデータを送りたい」

↓

ステップ2: PC-Aは「これは自分のネットワーク外だ」と判断

↓

ステップ3: パケットを「ルーターA」に送る

↓

ステップ4: ルーターAがインターネット経由でルーターBに転送

↓

ステップ5: ルーターBが自分のネットワーク内のPC-Bに配送

## ルーターの主な役割

1. 異なるネットワーク間の橋渡し
2. 家庭内ネットワーク ↔ インターネット
3. ネットワークA ↔ ネットワークB
4. パケットの転送 (ルーティング)
5. 宛先IPアドレスを見て、どこに送るべきか判断
6. 最適な経路を選択
7. **NAT**によるアドレス変換
8. プライベートIPとグローバルIPの変換
9. 複数の機器で1つのグローバルIPを共有



## なぜ必要なのか？

【シナリオ】PC-AがGoogle（8.8.8.8）にアクセスしたい

PC-Aの判断プロセス：

ステップ1：「8.8.8.8に接続したい」

↓

ステップ2：「8.8.8.8は192.168.1.xじゃない」

↓

ステップ3：「これは自分のネットワーク外だ！」

↓

ステップ4：「デフォルトゲートウェイ（192.168.1.1）に送ろう」

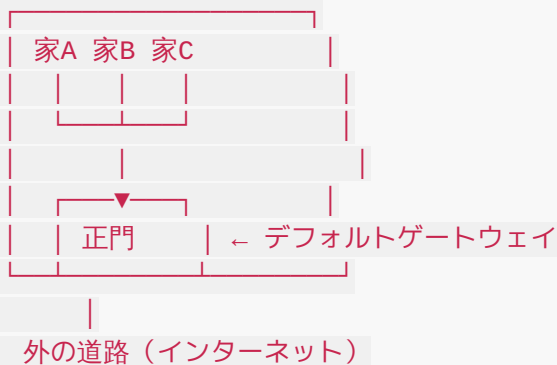
↓

ステップ5：ルーターが代わりにインターネットへ転送

## 道路の出口に例えると

【住宅地のネットワーク】

住宅街（プライベートネットワーク）



- 住宅街の中を移動 → 直接行ける
- 住宅街の外に行く → 必ず正門を通る

**まとめ：** - デフォルトゲートウェイ = 外部ネットワークへの出口 - 通常はルーターのIPアドレス  
- 「自分のネットワーク外」へのパケットは全てここに送る

## Q7: OSIモデルとは何ですか？

**A:** OSIモデルとは、ネットワーク通信を**7つの階層（レイヤー）**に分けて整理した概念モデルです。



Q2で学んだ「電気信号」から「データのやり取り」まで、複雑な処理を段階的に理解するための枠組みです。

## なぜ階層に分けるのか？

### 【料理に例えると】

料理を作る工程を分担：

5. 盛り付け担当	← 見た目を整える
4. 調理担当	← 焼く・煮る
3. 下ごしらえ担当	← 切る・洗う
2. 食材調達担当	← 買い物
1. メニュー企画担当	← 何を作るか決める

各担当は自分の役割だけに集中できる

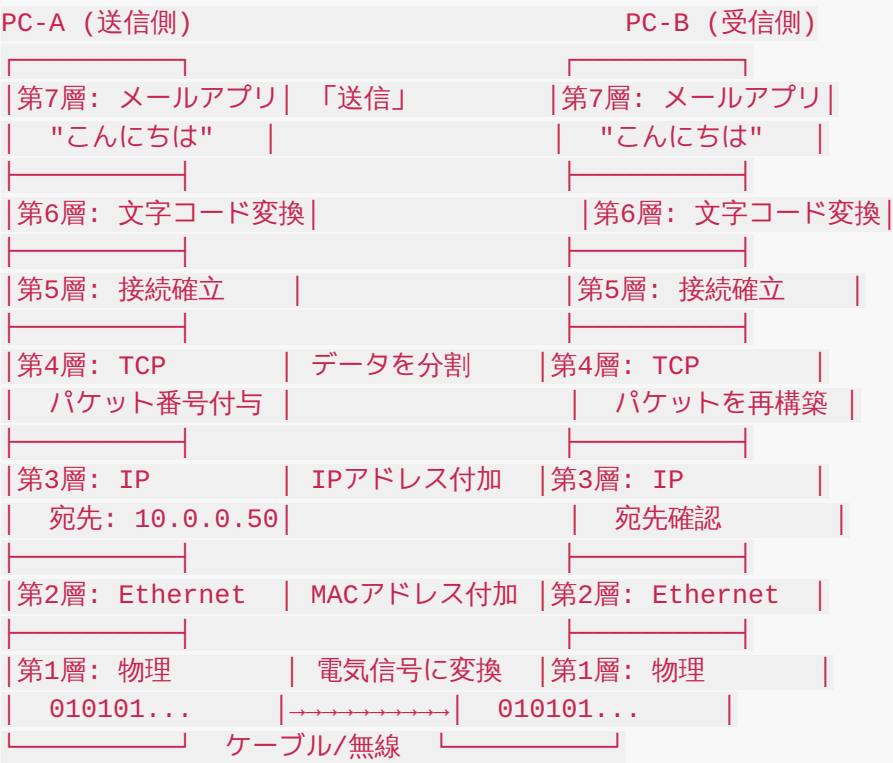
→ 効率的で、問題があれば該当層だけ修正すればいい

## OSI 7階層モデル

階層	名前	役割
第7層	アプリケーション層 (Application)	ユーザーが使うアプリ (Web, メール等)
第6層	プレゼンテーション層 (Presentation)	データ形式の変換 (暗号化、圧縮)
第5層	セッション層 (Session)	通信の開始・終了管理
第4層	トランスポート層 (Transport)	信頼性のある通信 (TCP, UDP)
第3層	ネットワーク層 (Network)	経路選択(ルーティング) (IP)
第2層	データリンク層 (Data Link)	直接繋がった機器間通信 (Ethernet)
第1層	物理層 (Physical)	電気信号、ケーブル (0と1の世界)

# 実際の通信の流れ

【PC-AからPC-Bにメールを送る例】



【流れ】

送信: 第7層→第6層→...→第1層 (カプセル化: データに情報を付加)  
受信: 第1層→第2層→...→第7層 (デカプセル化: 情報を取り出す)

## 重要な層の具体例

### 【第1層（物理層）】

0101010110101... | ← 電気信号、光信号

例：LANケーブル、光ファイバー

### 【第3層（ネットワーク層）】

送信元IP: 192.168.1.10 |

宛先IP: 10.0.0.50 |

データ... |

例：IPプロトコル、ルーター

### 【第7層（アプリケーション層）】

HTTPリクエスト |

GET /index.html |

例：Webブラウザ、メールソフト

**まとめ:** - OSIモデル = ネットワーク通信を7階層で整理 - 各階層が独立した役割を持つ - 下の層（物理的）から上の層（アプリ）へと段階的に処理

## Q8: Dockerのネットワークとローカルネットワークはどう繋がっているのですか？

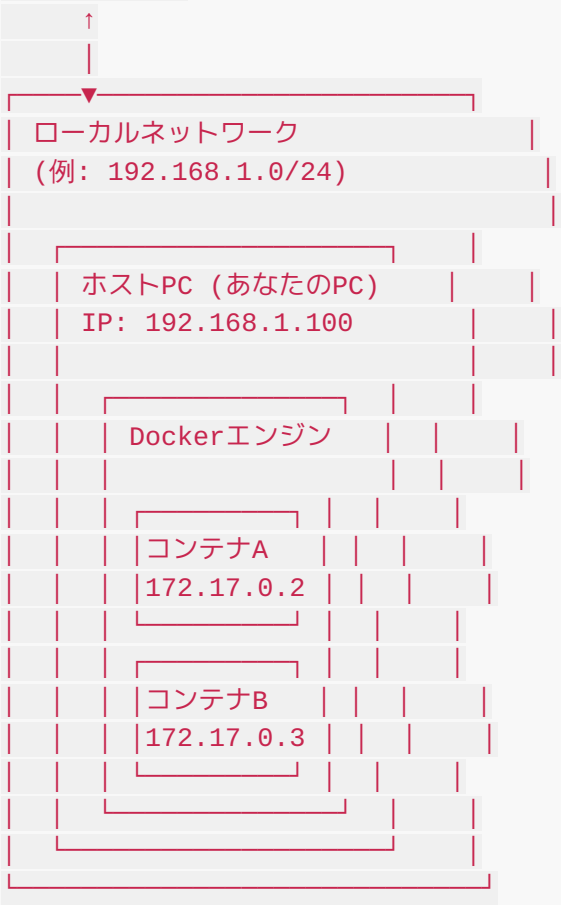
**A:** Dockerのネットワークは、**仮想的なネットワーク環境**で、ホストPC（あなたのPC）のネットワークを通じてローカルネットワークやインターネットに接続します。

これまで学んだ、IPアドレス（Q4）、ルーター（Q5）、デフォルトゲートウェイ（Q6）の知識を組み合わせると理解できます。

# 全体像

## 【ネットワークの階層構造】

インターネット

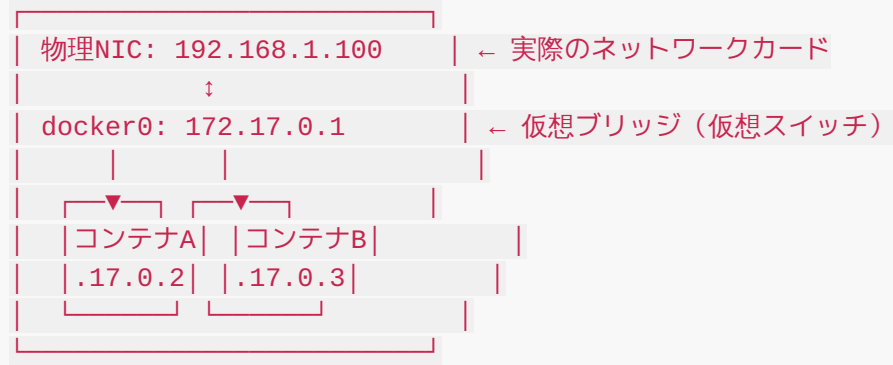


## Dockerネットワークの仕組み

### 【ステップ1】 Dockerがデフォルトで作るネットワーク

Dockerは起動時に「docker0」という仮想ブリッジを作成：

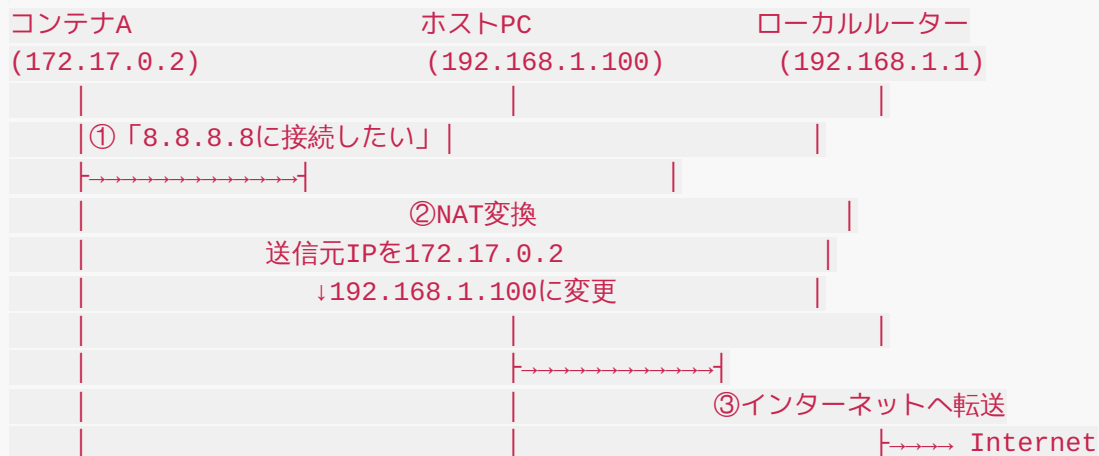
ホストPC



docker0 = 仮想的な「スイッチングハブ」の役割

## 通信の具体例

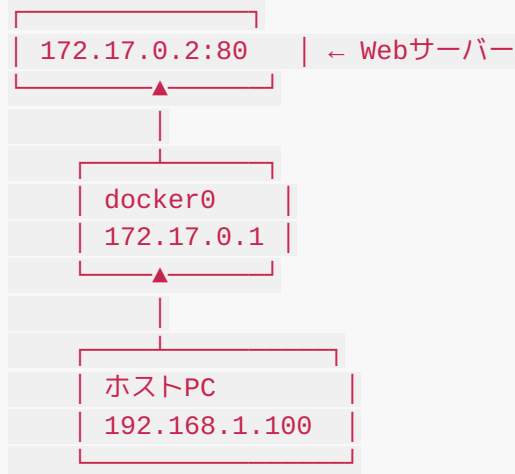
### 【例1】 コンテナAからインターネットにアクセス



NAT（ネットワークアドレス変換）： - Q5で学んだルーターのNATと同じ仕組み - Dockerもホスト内でNATを使う - コンテナのプライベートIP → ホストのIPに変換

### 【例2】ホストPCからコンテナのWebサーバーにアクセス

コンテナA (Nginxが80番ポートで起動)

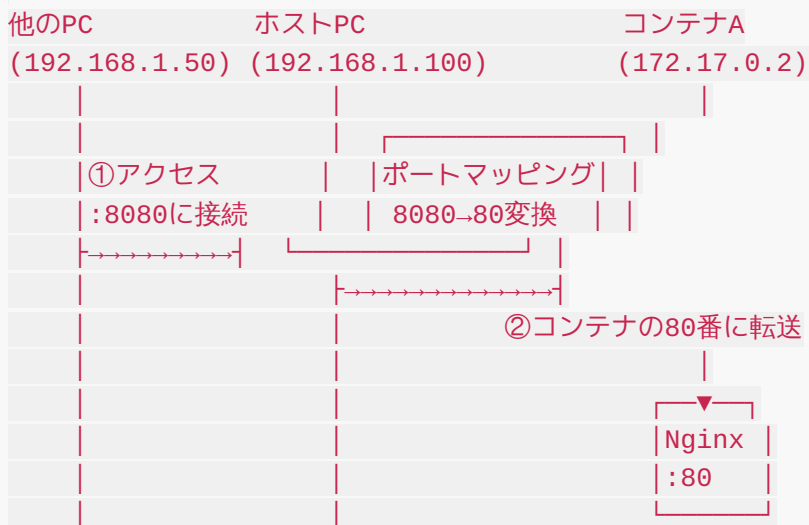


ホストから: `http://172.17.0.2:80`

→ 直接コンテナにアクセス可能

### 【例3】ローカルネットワークの他のPCからコンテナにアクセス

ポートフォワーディング (ポートマッピング) を使用:



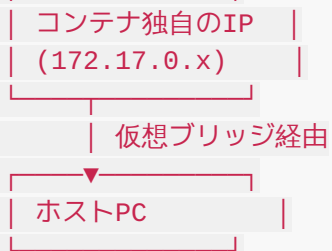
起動コマンド例:

```
docker run -p 8080:80 nginx
```

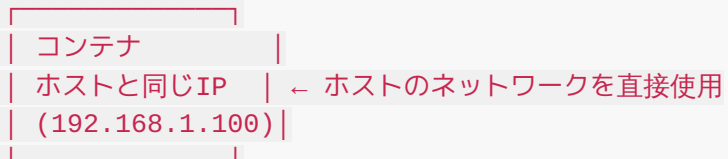
-p 8080:80 = ホストの8080番ポート → コンテナの80番ポートに転送

## Dockerネットワークモード

### 1. ブリッジモード（デフォルト）



### 2. ホストモード



### 3. なしモード





## ローカルネットワークスキャンツールとの関係

### 【このLocalNetScanツールの動作】

コンテナ内で実行する場合：

Dockerコンテナ

LocalNetScanツール

192.168.1.0/24を

スキャンしたい

デフォルトだと172.17.0.xから

スキャンすることになる

ホストモードで起動すれば：

▼  
ホストPCのネットワークを直接使用

192.168.1.100から

192.168.1.0/24をスキャン

起動コマンド：

```
docker run --network host localnet scan
```

**重要なポイント：** 1. Dockerコンテナは仮想的な独立ネットワークを持つ 2. デフォルトでは172.17.0.x のIPアドレスが割り当てられる 3. ホストPCのネットワーク機能（NAT）を通じて外部と通信 4. ポートマッピングで外部からコンテナにアクセス可能 5. ホストモードを使えばホストと同じネットワークを使用可能

# まとめ: すべての概念を繋げる

---

## 学んだことの全体像

### 【階層的な理解】

#### レベル1: 物理的な接続

- ト 電気信号がケーブルを通じて伝わる (Q2)
- ト 物理層 (OSI第1層) (Q7)
- ↳ これが「繋がる」の土台

#### レベル2: データの解釈

- ト 電気信号をパケットとして解釈 (Q2)
- ト IPアドレスで宛先を識別 (Q4)
- ト データリンク層、ネットワーク層 (Q7)
- ↳ 「どこに」「何を」送るかを理解

#### レベル3: ネットワーク間の接続

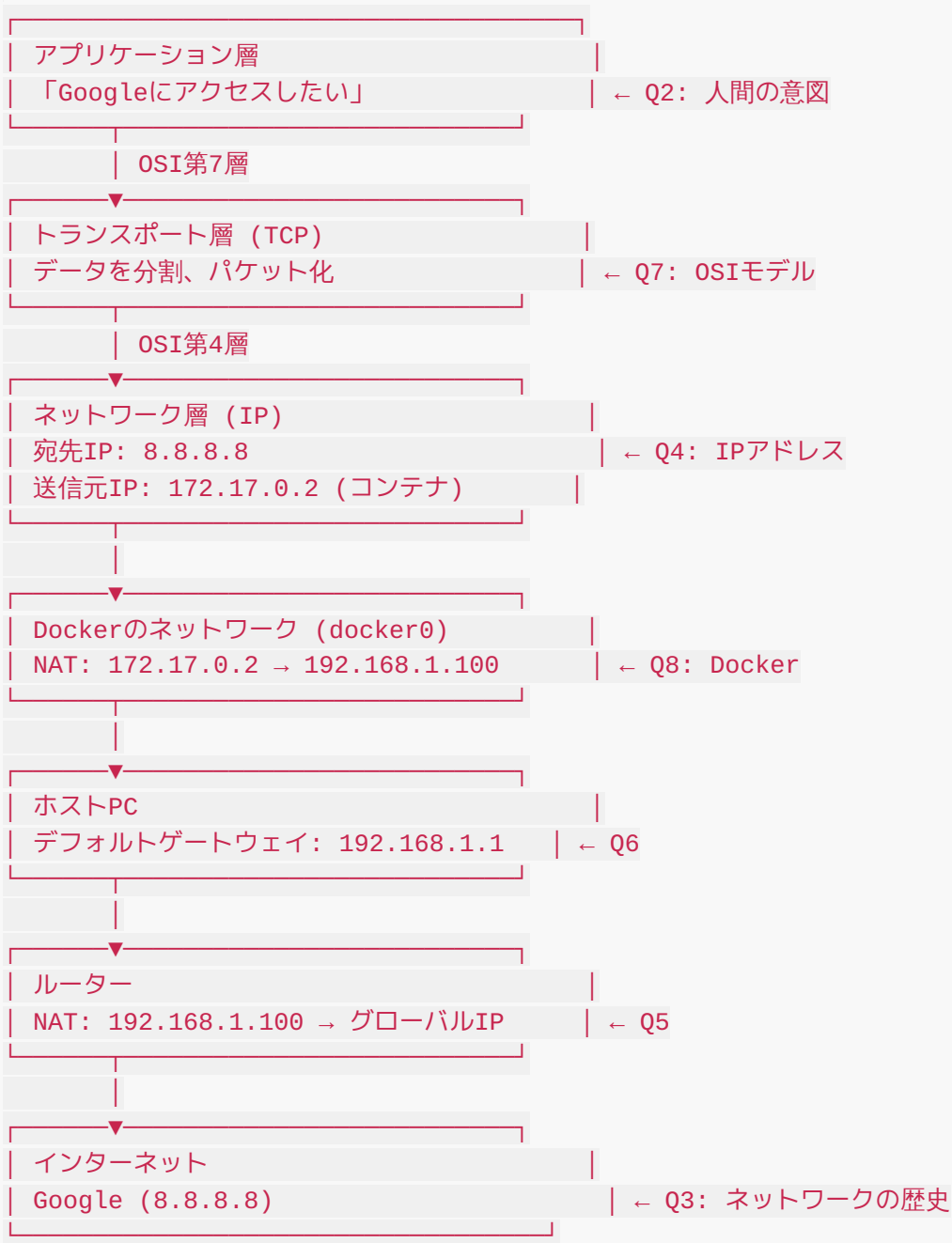
- ト ルーターが異なるネットワークを繋ぐ (Q5)
- ト デフォルトゲートウェイで外に出る (Q6)
- ↳ 「自分の家」から「外の世界」へ

#### レベル4: 仮想化環境

- ト Dockerコンテナの仮想ネットワーク (Q8)
- ト ホストPCを経由して実ネットワークに接続
- ↳ 「入れ子構造」のネットワーク

## 最終的な通信フロー図

### 【コンテナ内のアプリからインターネットまで】



# 人間のコミュニケーションとの最終比較

人間		コンピュータ	
話したいことがある	≡	送信データがある	
日本語で表現	≡	プロトコルで表現	
音声に変換	≡	パケットに変換	
空気中を音波が伝わる	≡	電気信号が伝わる	
相手の耳で受信	≡	NICで受信	
脳で言葉を理解	≡	パケットを解釈	
返事をする	≡	応答パケットを返す	
重要：どちらも「共通のルール」が必要			
人間 → 言語			
コンピュータ → プロトコル (TCP/IP等)			

## キーワード集

キーワード	意味
ネットワーク	複数のコンピュータが接続して通信する仕組み
電気信号	コンピュータが通信に使う0と1のデジタル信号
パケット	ネットワークで送られるデータの単位（宛先情報付き）
プロトコル	通信のルール（人間の「言語」に相当）
IPアドレス	ネットワーク上の住所（例: 192.168.1.10）
ルーター	異なるネットワーク同士を繋ぐ装置
デフォルトゲートウェイ	外部ネットワークへの出口（通常はルーターのIP）
NAT	プライベートIPとグローバルIPを変換する技術
OSIモデル	ネットワーク通信を7階層に分けた概念モデル
TCP/IP	インターネットで使われる標準プロトコル
Docker	アプリを仮想環境（コンテナ）で動かす技術
ブリッジネットワーク	Dockerのデフォルトネットワークモード
ポートマッピング	ホストのポートをコンテナのポートに転送する設定

## さらに学ぶために

1. 実践してみよう
2. `ping` コマンドで他のPCに到達できるか確認
3. `ipconfig` (Windows) や `ip addr` (Linux) で自分のIPアドレスを確認
4. `tracert` でパケットの経路を試みる
5. このリポジトリのツールを使おう
6. LocalNetScanでネットワーク内の機器を発見

7. 実際にどんなIPアドレスの機器があるか確認
  8. 次のステップ
  9. サブネットマスクとは？
  10. DNSの仕組み (IPアドレス ↔ ドメイン名の変換)
  11. ファイアウォールとセキュリティ
  12. VLANによるネットワーク分割
- 

作成日: 2025年11月15日 対象: ネットワーク初学者 リポジトリ: LocalNetScan