

研究室向けの仮想化基盤システムの構築

坂東 恭幸

香川大学大学院

はじめに

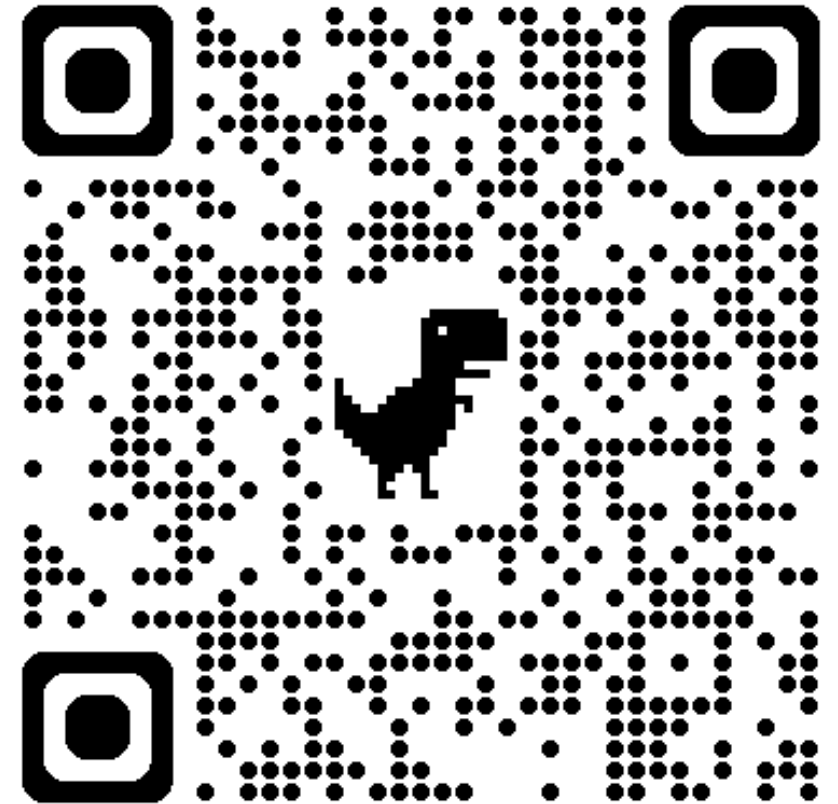
■ 自己紹介

- 坂東 恭幸 (ばんどう つかさ)
- 香川大学 大学院1年
- 喜田研究室、SLP（学生プログラミング研究所）所属
- サーバ・ネットワークを専攻

■ 本日のテーマ

- 研究室向けの仮想化基盤システムの構築
- 利用者
 - 非インフラ系の研究室の学生
- 利用用途
 - 研究（機械学習、セキュリティ）
 - 研究室HPの公開
 - 個人・チームで開発したアプリの公開 などなど

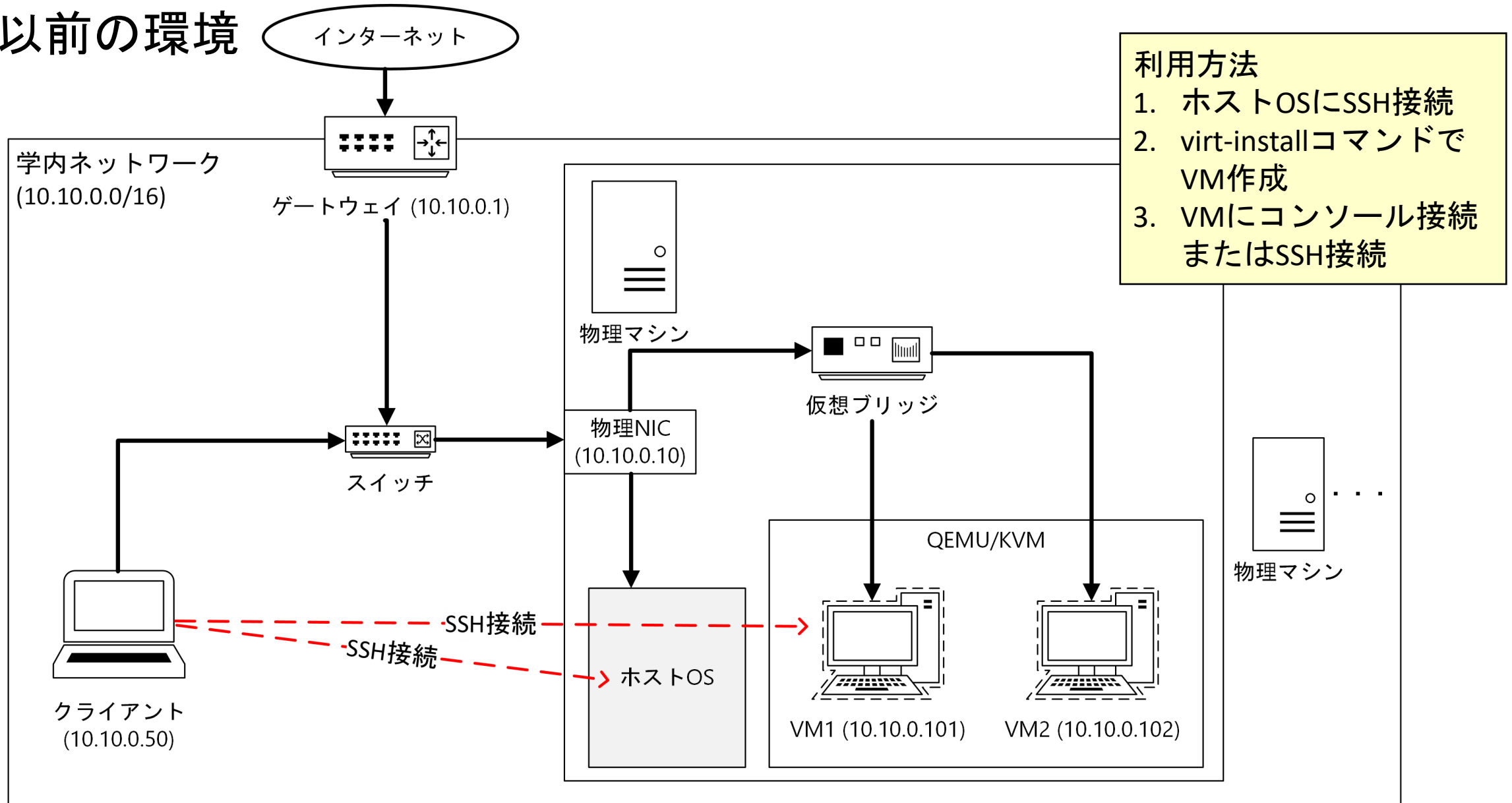
本日の資料は以下のQRコードから
ご覧いただけます



開発背景

開発背景

■ 以前の環境



開発背景

■ 問題点

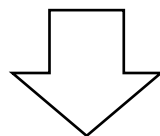
- 非インフラ系の学生にとって、コマンドを用いたVMの作成・管理が困難
- VMの定期バックアップを行っておらず、突然の物理サーバの故障などが原因でVM内の研究データが消失する恐れ
- 特定のサーバに負荷が集中した際、VMを移行する手段が複雑であり、研究室全体でリソースを効率的に利用できない
- 物理サーバごとにOSのバージョンや設定がバラバラであり、トラブル発生時に原因究明が困難
- ネットワーク構成が「研究室向け」には不適

解決策

解決策

■ コンセプト

利用者がVMを自由に作成・管理できる仮想化基盤システム



Proxmoxを採用

■ Proxmoxとは？

- 仮想マシンとコンテナの両方を統合管理する仮想化プラットフォーム
- 最大の決め手：Webインタフェース
 - ブラウザ上でVMの作成、起動、停止、コンソール操作まで完結
 - バックアップ、クラスタ、仮想NW・FWの管理も可能
 - VMのリソースモニターがわかりやすい



解決策

■ コマンド VS Webインタフェース

```
$ virt-install ¥  
  --name ubuntu22-server ¥  
  --memory 2048 ¥  
  --vcpus 2 ¥  
  --disk path=/var/lib/libvirt/images/ubuntu22-  
server.qcow2,size=20,format=qcow2 ¥  
  --os-variant ubuntu22.04 ¥  
  --network network=default ¥  
  --graphics none ¥  
  --console pty,target_type=serial ¥  
  --location  
'http://jp.archive.ubuntu.com/ubuntu/dists/ja  
mmy/main/installer-amd64/' ¥  
  --extra-args 'console=ttyS0,115200n8 serial'
```

作成: 仮想マシン

全般 OS システム ディスク CPU メモリ ネットワーク 確認

ノード: fugu リソースプール:

VM ID: 1005

名前: test-vm

? ヘルプ 詳細設定 ☐ 戻る 次へ

解決策

■ Proxmoxの機能・特徴による問題点解決

- ・ 非インフラ系の学生にとって、コマンドを用いたVMの作成・管理が困難

⇒ WebインタフェースからVMを作成・管理

- ・ VMの定期バックアップを行っておらず、突然の物理サーバの故障などが原因でVM内の研究データが消失する恐れ

⇒ Proxmox Backup Serverを使った定期的なバックアップ

- ・ 特定の物理サーバに負荷が集中した際、VMを移行する手段が複雑であり、研究室全体でのリソース利用が非効率的

⇒ VMを別の物理サーバへ移行（マイグレーション機能）

- ・ 物理サーバごとにOSのバージョンや設定がバラバラであり、トラブル発生時に原因究明が困難

⇒ クラスタ構築による、一元管理

ここまでは
Proxmoxで解決

-
- ・ ネットワーク構成が「研究室向け」には不適

- ・ VMが学内ネットワークに配置され、学内のIPリソースを圧迫する
- ・ すべてのVMが同じネットワークに存在するため、他人のVMにセキュリティ的な影響を及ぼす

改めて「研究室向け」とは？

「研究室向け」とは？

■ 利用者は...?

- 非インフラ系の研究室の学生

⇒ 複数人が利用、インフラのプロではない

■ 利用用途は...?

- 研究（セキュリティ）

⇒ 攻撃ツールを含むソフトウェアをインストール

- 個人・チームで開発したアプリの公開

⇒ Webサーバ + DBサーバの構築



重要

セキュリティの確保と利便性向上の両立
なぜ重要なのか？

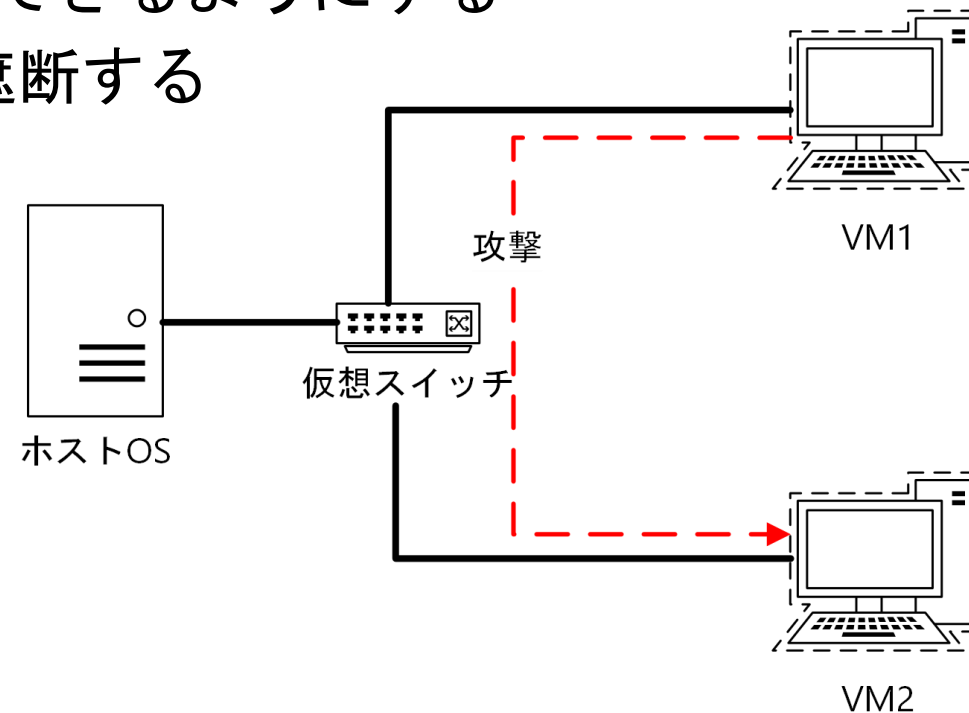
例えばこんなシチュエーションがある

■ シチュエーション①

- セキュリティの研究をしている学生
 - 研究用に攻撃ツールをインストールしたい
 - 誤った操作によって、他人のVMに攻撃を行う可能性がある

■ 要件①

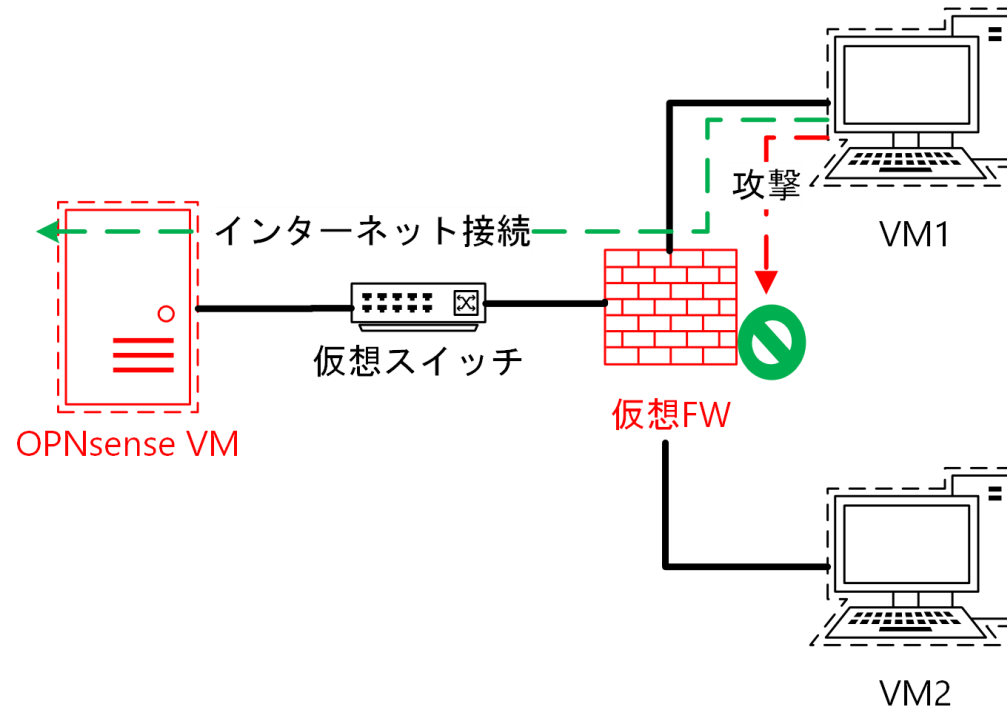
- インターネットに接続できるようにする
- 他VMへの通信を一切遮断する



例えばこんなシチュエーションがある

■ 解決手法①

- ProxmoxのSDN機能でVM共通のインターネット接続用の仮想ネットワークを作成し、**仮想ファイアウォール**で他VMとの通信を遮断するルールを作成
- ホストの代わりに**ルーティングソフトウェア(OPNsense)のVM**を設置し、学内ネットワークに接続
 - VMから学内ネットワークへの通信はNAPT変換されることで、学内のIPリソースを節約



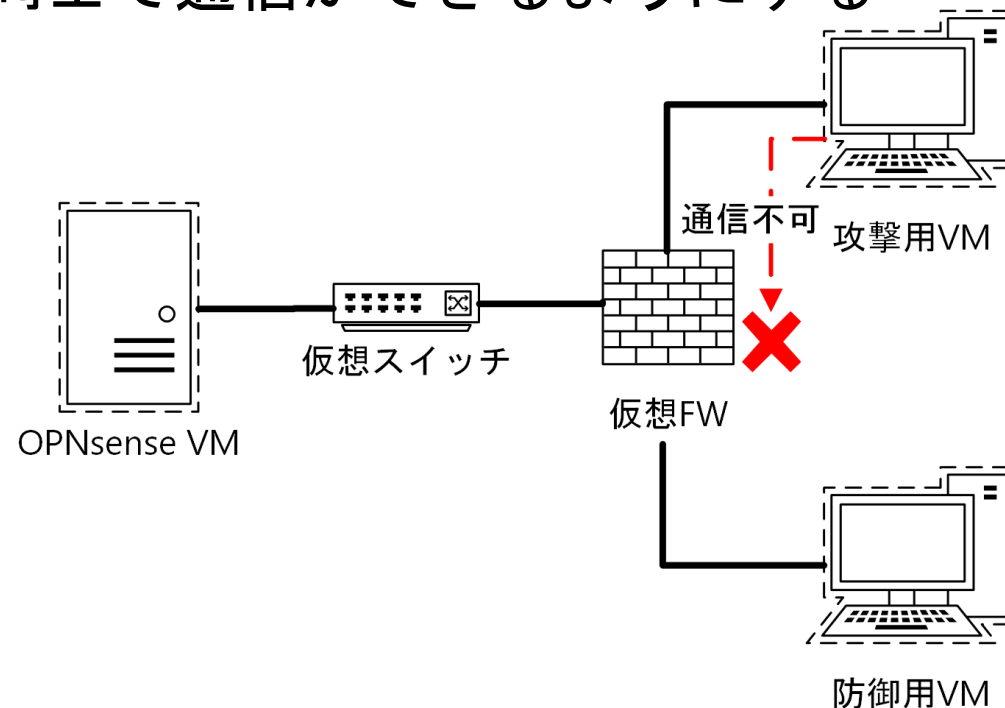
例えばこんなシチュエーションがある

■ シチュエーション②

- セキュリティの研究をしている学生
 - 攻撃ツールを使って対象のVMに攻撃を行い、攻撃による影響を確認したい
- Webサーバ + DBサーバのVMと連携して、サービスを提供したい

■ 要件②

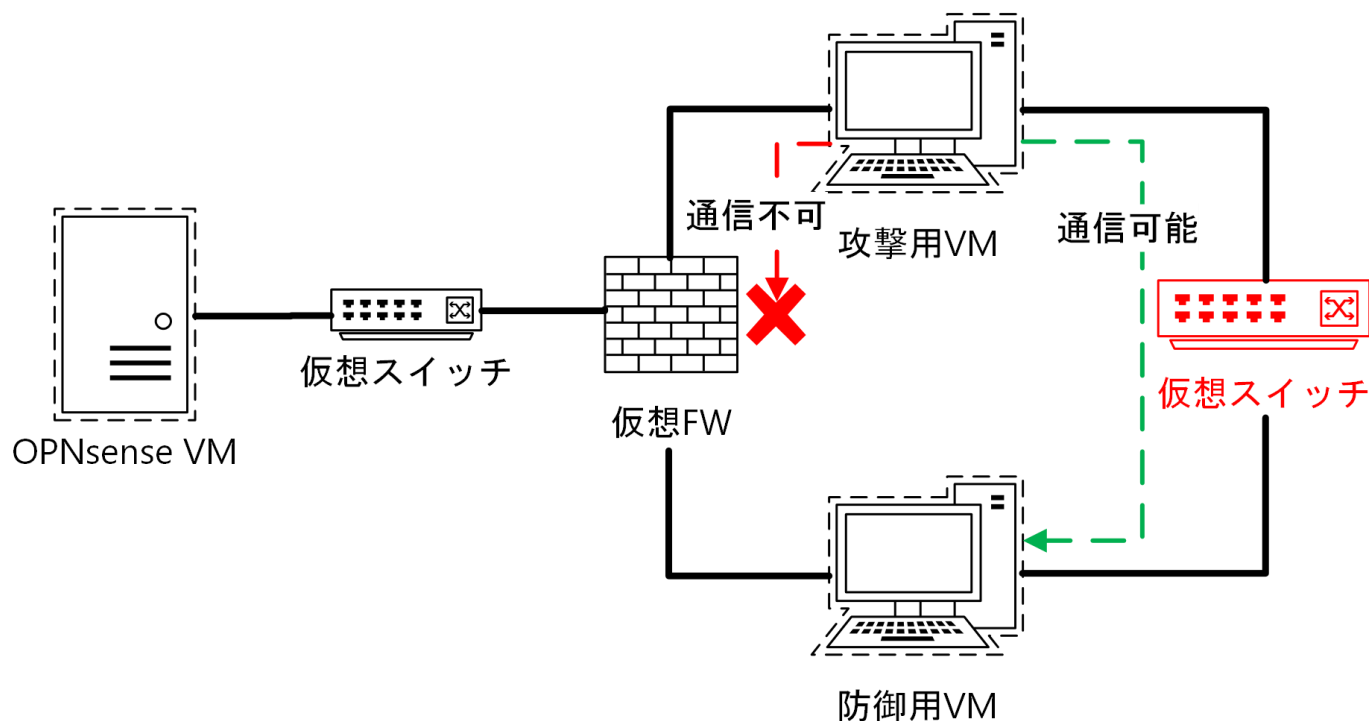
- ユーザが指定したVM同士で通信ができるようにする



例えばこんなシチュエーションがある

■ 解決手法②

- ProxmoxのSDN機能で各ユーザに**仮想ネットワーク（ユーザネットワーク）**を提供
 - ユーザは指定したVMにユーザネットワーク用のNICを追加可能
 - ユーザは用途別のVMにVLANタグをつけることが可能



システム構成図

システム構成図

学内ネットワーク

利用者: 8人
VM: 20台
物理マシン: 10台
スイッチ: 2台
NW速度: 1Gbps

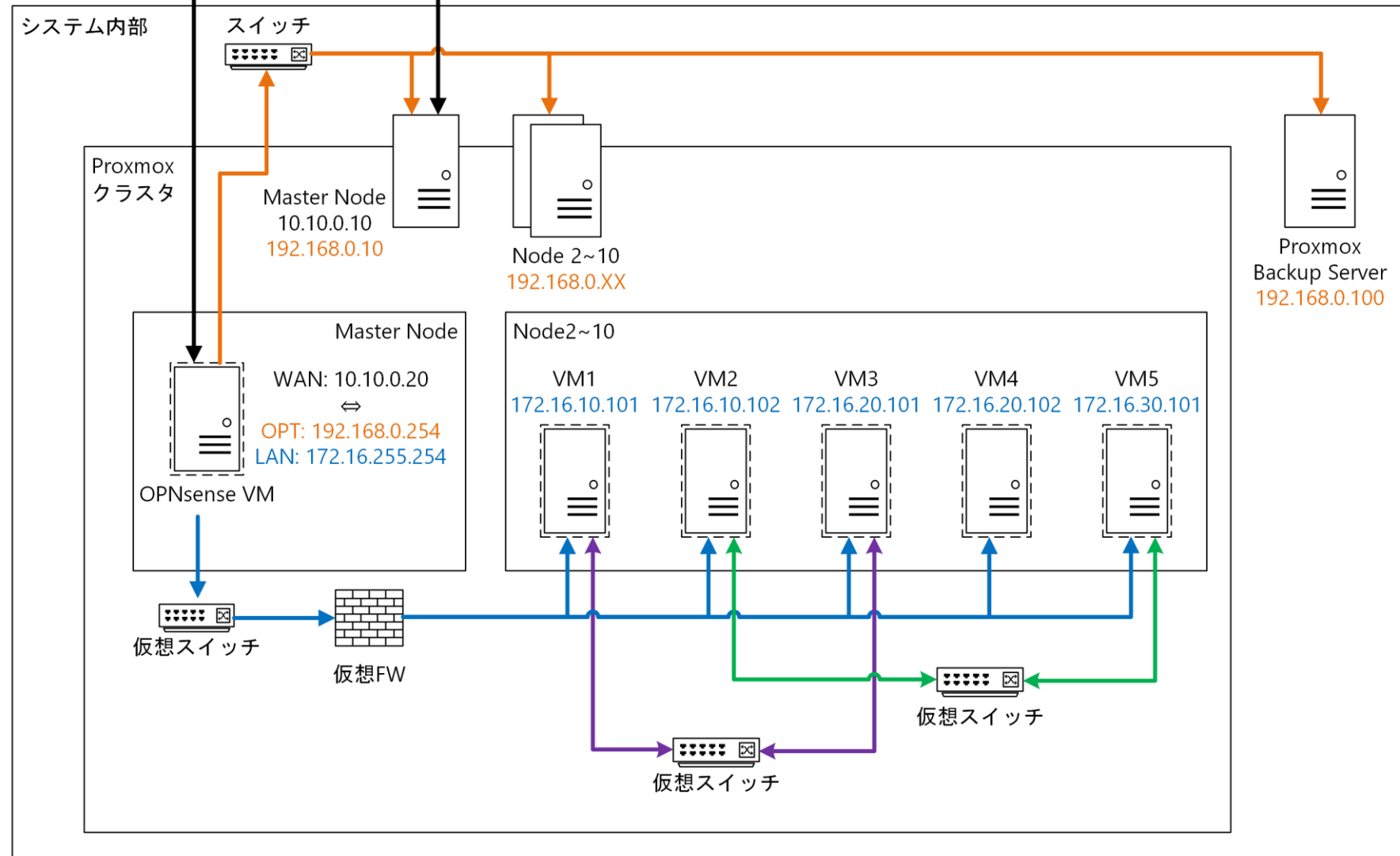
学内ネットワーク
10.10.0.0/16

Proxmox クラスタ
ネットワーク
192.168.0.0/24

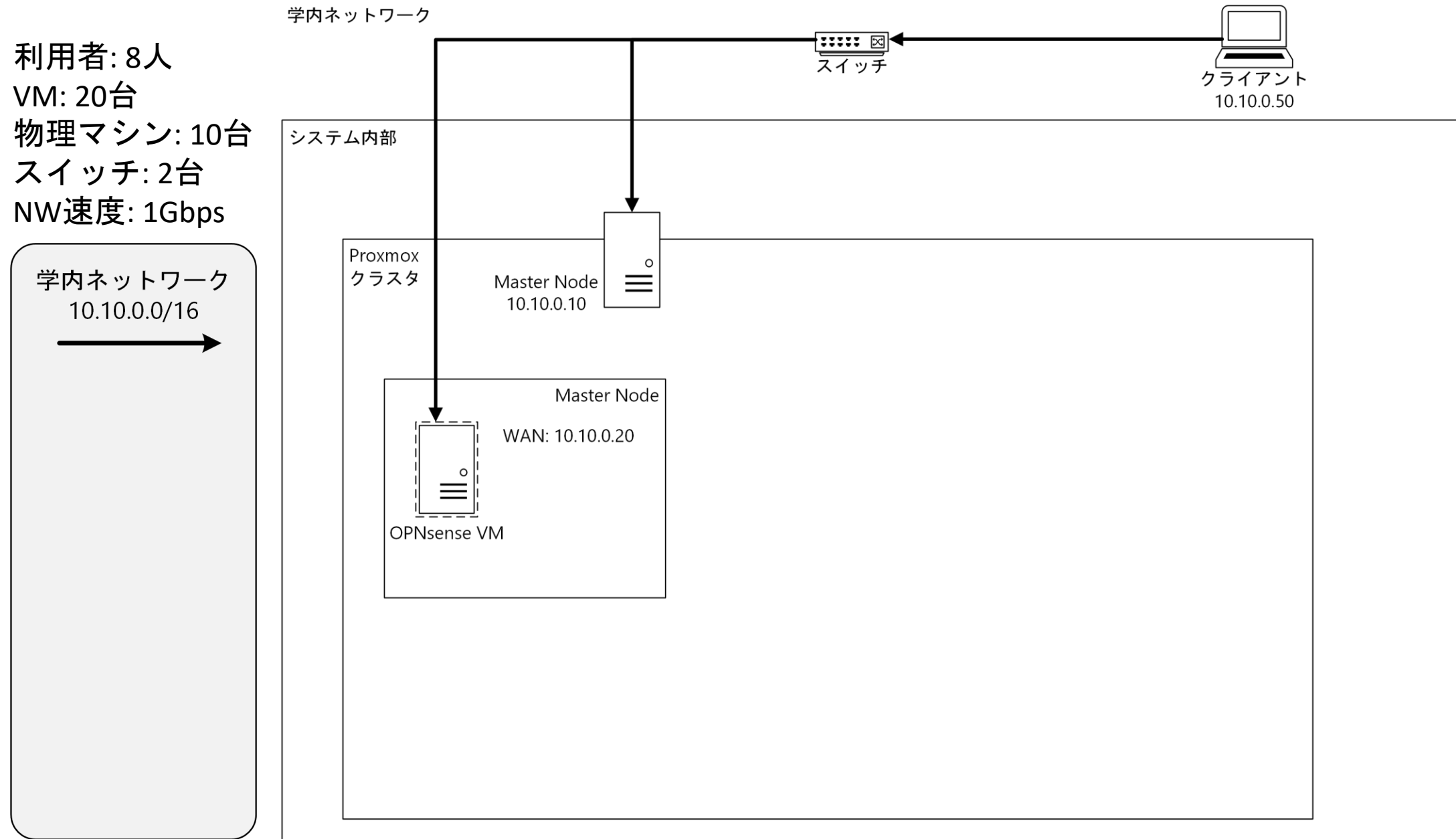
VM共通
ネットワーク
172.16.0.0/16

ユーザネットワーク
(User A)

ユーザネットワーク
(User B)



システム構成図



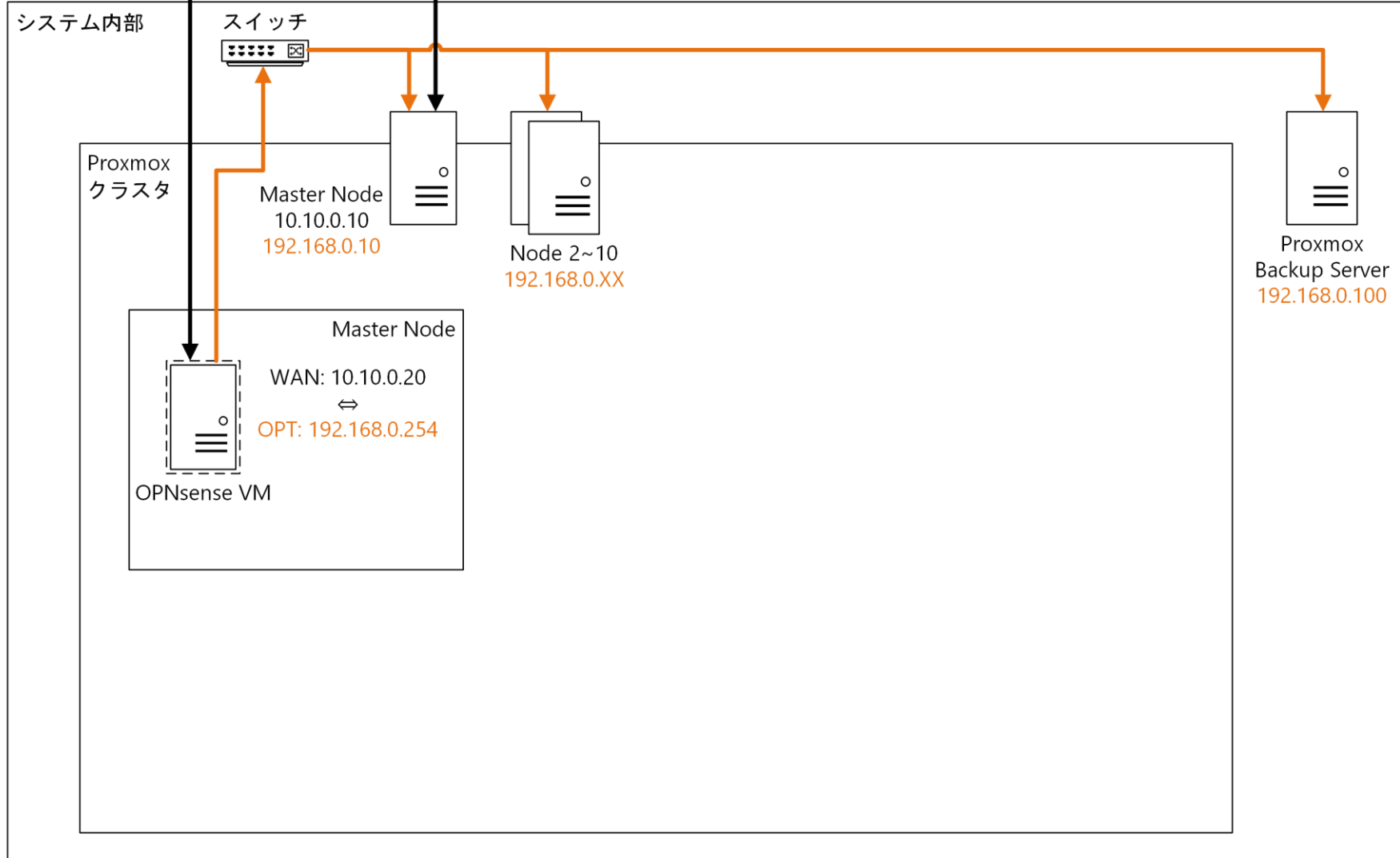
システム構成図

学内ネットワーク

利用者: 8人
VM: 20台
物理マシン: 10台
スイッチ: 2台
NW速度: 1Gbps

学内ネットワーク
10.10.0.0/16

Proxmox クラスタ
ネットワーク
192.168.0.0/24



システム構成図

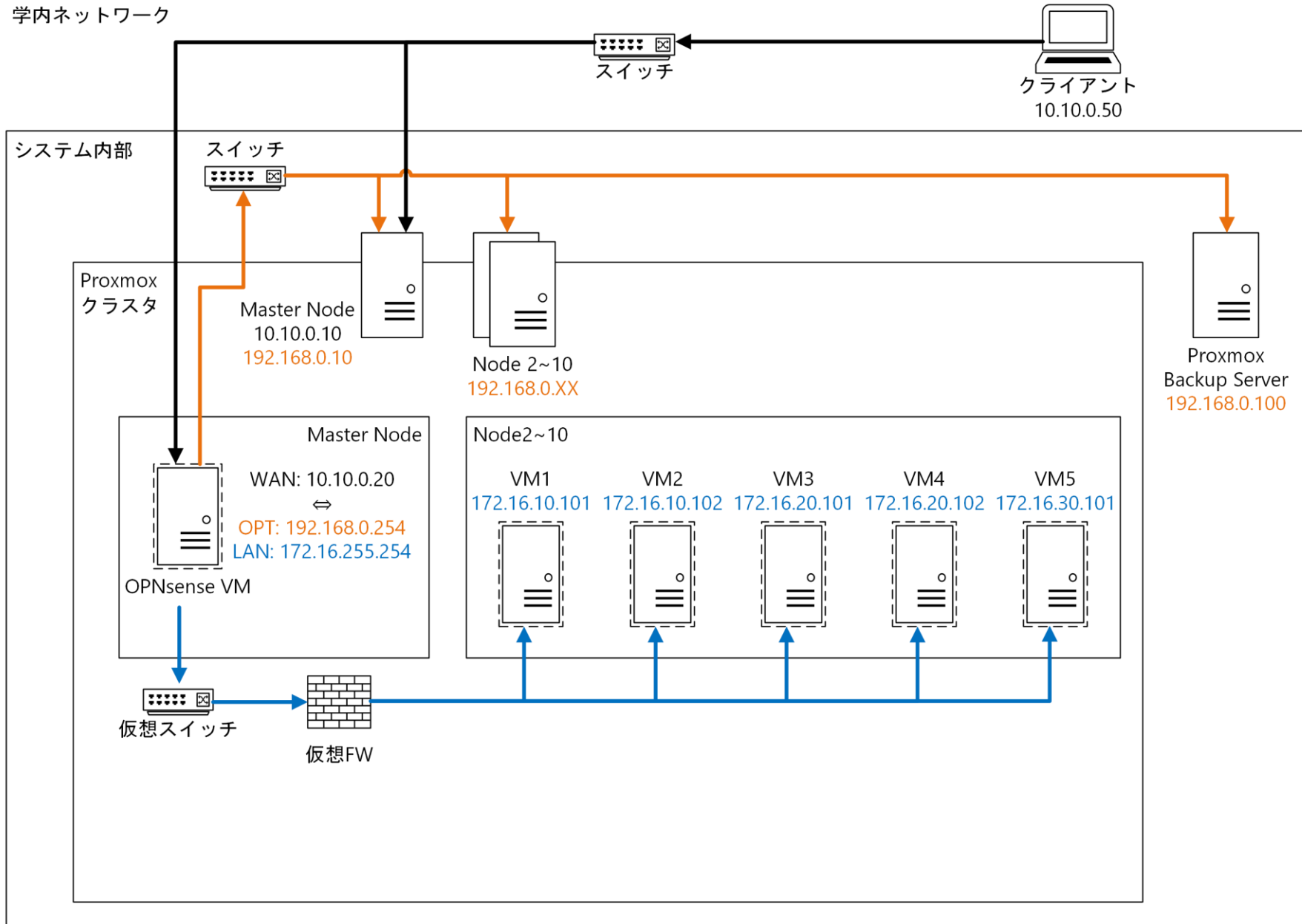
利用者: 8人
VM: 20台
物理マシン: 10台
スイッチ: 2台
NW速度: 1Gbps

学内ネットワーク
10.10.0.0/16

Proxmox クラスタ
ネットワーク
192.168.0.0/24

VM共通
ネットワーク
172.16.0.0/16

学内ネットワーク



システム構成図

学内ネットワーク

利用者: 8人
VM: 20台
物理マシン: 10台
スイッチ: 2台
NW速度: 1Gbps

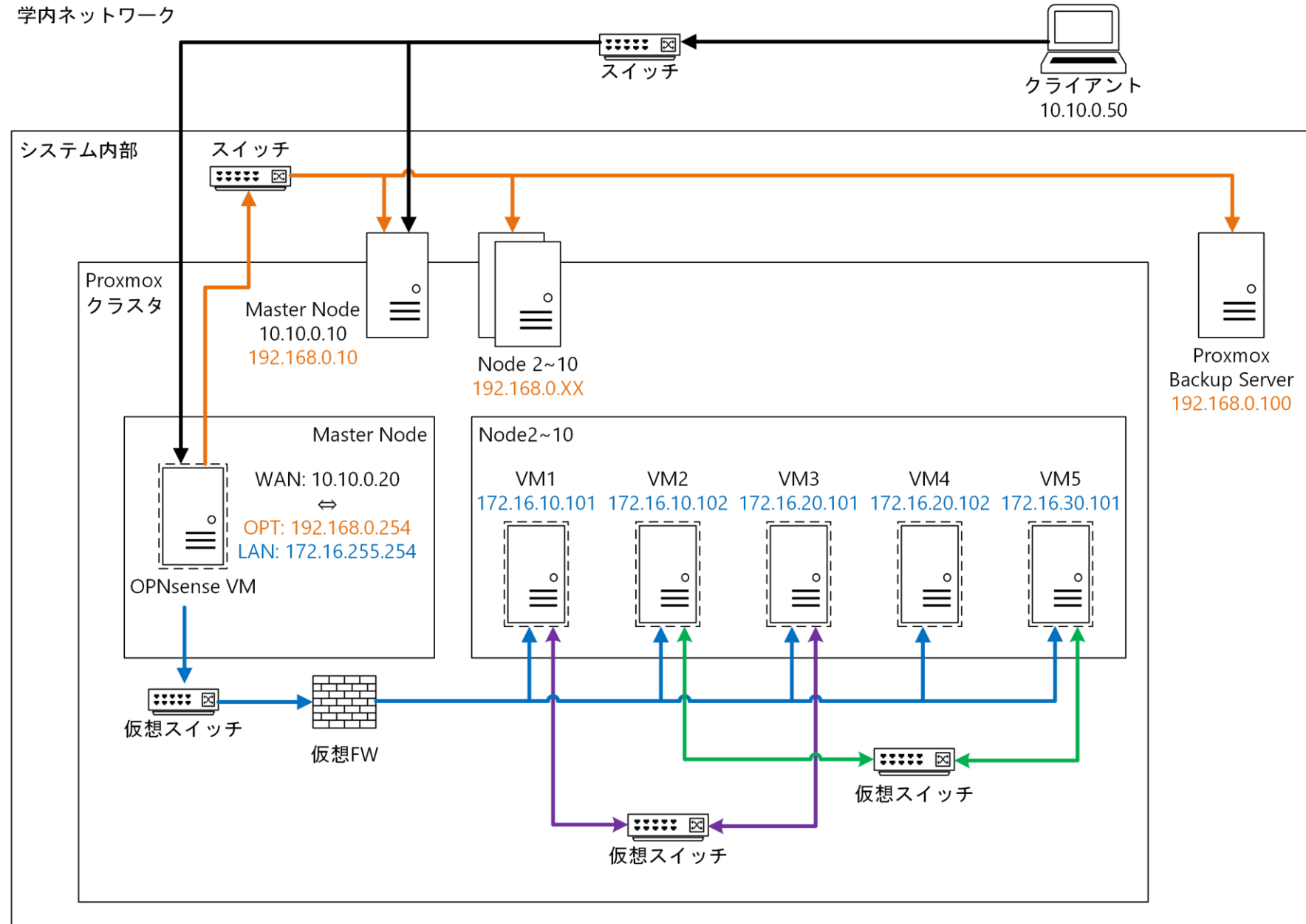
学内ネットワーク
10.10.0.0/16

Proxmox クラスタ
ネットワーク
192.168.0.0/24

VM共通
ネットワーク
172.16.0.0/16

ユーザネットワーク
(User A)

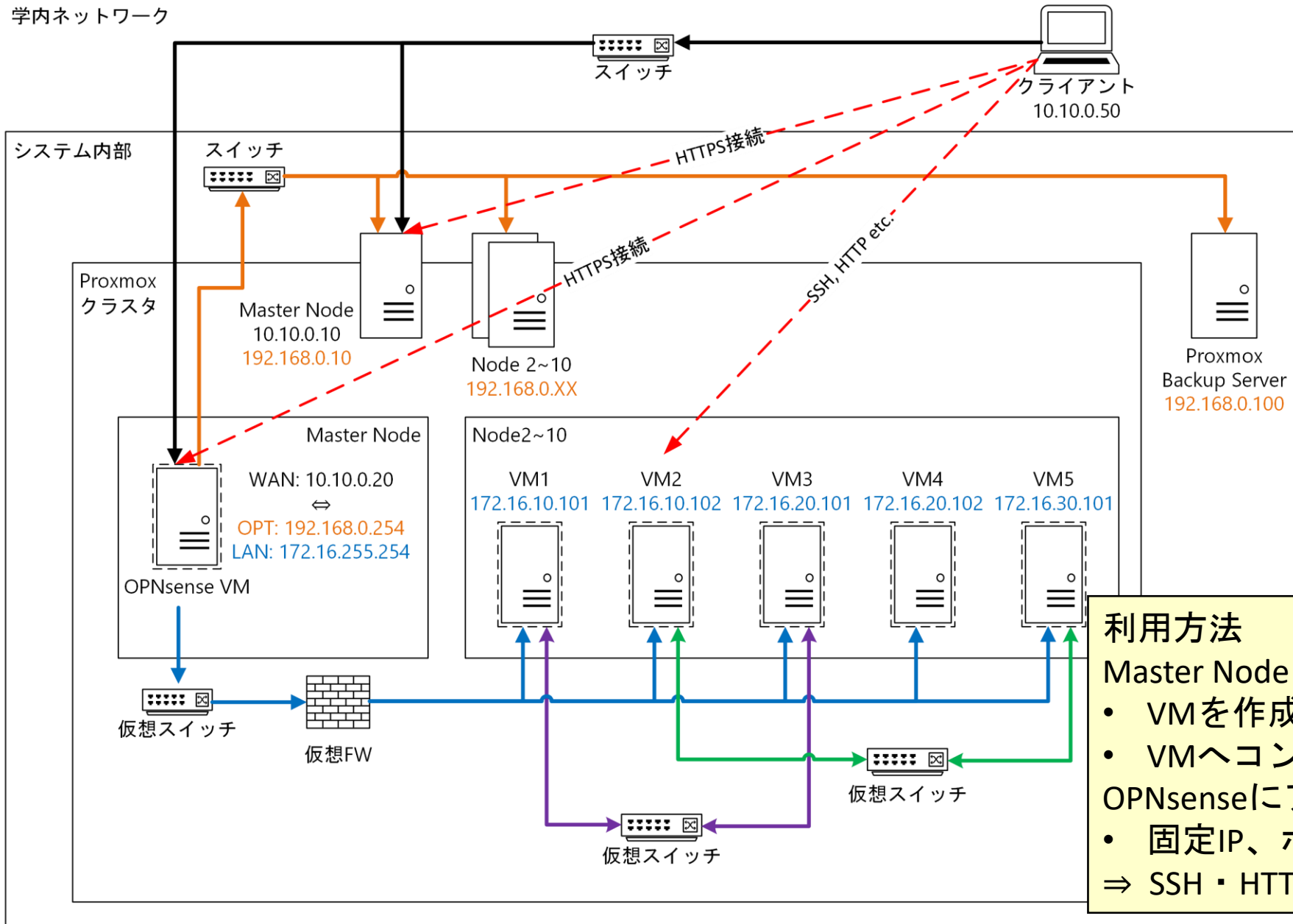
ユーザネットワーク
(User B)



システム構成図

学内ネットワーク

利用者: 8人
VM: 20台
物理マシン: 10台
スイッチ: 2台
NW速度: 1Gbps



利用方法

- Master Nodeにアクセス
 - VMを作成
 - VMへコンソール接続
- OPNsenseにアクセス
 - 固定IP、ポートマッピング

⇒ SSH・HTTP接続が可能

利用者からの評判

利用者からの評判

- VMの作成がスピーディー
- VM作成のハードルが低い
- 物理マシンを管理しなくていい
- マイグレーション機能ありがたい
- WindowsのVMが使えるのが助かる
- ネットワークの設定がとても楽になった

⇒ おおむね好評！

今後の展望

今後の展望

■ 実運用して気づいたこと








- ProxmoxのWebインターフェースによって、VMの作成・管理が簡単にはなったが...
 - ドキュメント凝視しないとなかなか使うのは難しい
 - とっつきやすいデザインとはいえない
 - ヘルプを呼ばれることがあった

⇒ ProxmoxのAPIを使ってVMを管理できる **モダンなWebインターフェース**の開発

■ その他

- 認証基盤、監視ツールの導入
- ネットワークの冗長化

仮想マシン一覧

タイプ	ステータス	名前 / VMID	ノード	リソース使用量(Demo)	操作	詳細設定
 仮想マシン	 稼働中	fugu (1002)	salmon	CPU (0%) MEM (0%)	シャットダウン	コンソール
 仮想マシン	 稼働中	kaziki (101)	maguro	CPU (0%) MEM (0%)	シャットダウン	コンソール
 仮想マシン	 停止中	kanimiso (102)	kani	CPU (0%) MEM (0%)	起動	コンソール
 コンテナ	 停止中	CT (103)	maguro	CPU (0%) MEM (0%)	起動	コンソール

まとめ

■ 本日のテーマ

研究室向けの仮想化基盤システムの構築

■ ポイント

- 利用者が使いやすい

⇒ Webインタフェースが充実しているProxmoxを採用

- 「研究室向け」に必要な要件

⇒ 複数人が安全に利用できるようなネットワーク設計

- 仮想NW・FWやルーティングソフトウェアを用いたセキュリティの確保
- ユーザネットワークを用いた他VMとの通信による利便性向上