

# HCCJP 第63回勉強会

## ハイブリッドクラウド研究 会

2025年7月11日（金）14:00開始



# 本日のテーマ

## Azure + 生成AI

- Claude Code×Azure
- Gemini CLI×Azure
- Microsoft "Adaptive Cloud" 最新動向
- Azure Arc対応Kubernetes拡張機能 "Edge RAG"

## 自己紹介

胡田 昌彦（えびすだ まさひこ）

## 現在の活動

- YouTubeに注力中！ → <https://youtube.com/@ebibibi>

## 資格・認定

- Microsoft MVP for Azure Hybrid & Windows Server（ダブル受賞）

## 著書

- 「Windowsインフラ管理者入門」 著者

## 趣味

- ベース、ドラム、セッション



# このスライドについて

## 生成AIで作成しました！

### 使用技術

- Claude - スライド内容の生成
- Marp - Markdownベースのプレゼンテーション
- VS Code - 統合開発環境

今日のテーマにぴったり！生成AIの実践例です

# タイムテーブル

| 時刻    | 時間  | セッション                                    | スピーカー  |
|-------|-----|--|--------|
| 14:00 | 5分  | オープニング                                   | 胡田 昌彦  |
| 14:05 | 20分 | 【前半】 Claude Code×Azure, Gemini CLI×Azure | 胡田 昌彦  |
| 14:25 | 20分 | Microsoft "Adaptive Cloud" Updates       | 高添 修 氏 |
| 14:45 | 20分 | Azure Arc対応 K8s拡張機能 "Edge RAG"           | 胡田 昌彦  |
| 15:05 | 10分 | 【後半】 Claude Code×Azure, Gemini CLI×Azure | 胡田 昌彦  |
| 15:15 | 10分 | Q&A                                      | 全員     |
| 15:25 | 5分  | クロージング                                   | 胡田 昌彦  |

# オープニング

司会：胡田 昌彦

日本ビジネスシステムズ株式会社

Microsoft MVP for Azure Hybrid, Windows  
Server



# HCCJPとは

## ハイブリッドクラウド研究会

- 毎月第2金曜日 14時から開催
- Azure + 生成AI関連の最新情報をお届け
- オンライン配信（YouTube HCCJPチャンネル）

📺チャンネル登録お願いします！

<https://www.youtube.com/channel/UCrf4bEI7yJnkGYo3F67gA7w>



# 本日の注意事項

- 📺 配信は録画されています→アーカイブでもいつでも見られます！
- 💬 チャットでの質問大歓迎！
- 📝 Q&Aセッションで匿名質問も可能！



# 質問方法

💡 Slidoで匿名質問ができます！

- 匿名で質問可能 - お気軽にどうぞ！
- いつでも質問OK - セッション中でも遠慮なく
- 疑問点はすぐに - 思いついたらすぐ投稿

## 📱 Slidoの使い方

1. QRコードまたはURLからアクセス
2. 質問を入力して送信
3. 他の方の質問に「いいね」も可能

**セッション中、疑問点があればいつでも質問してください！**


# セッション①【前半】

## Claude Code × Azure, Gemini CLI × Azure

スピーカー：胡田 昌彦

日本ビジネスシステムズ株式会社

Microsoft MVP for Azure Hybrid, Windows Server

 14:05 - 14:25 ( 20分 )

# セッション①について

## 本セッションのポイント

- 話題沸騰中のClaude CodeとGemini CLI
- Azure環境構築のTipsと実演
- **無料でも実現可能！**

コンソールベースの生成AIでまるで魔法のような体験を！

## 🤔 なぜCLIツールを選ぶのか？

### GUI vs CLI AIコーディングエージェント

| ツール            | タイプ       | 料金         |
|----------------|-----------|------------|
| Cline          | VS Code拡張 | 無料（API料金別） |
| GitHub Copilot | VS Code統合 | 無料～        |
| Cursor         | 専用エディタ    | 無料～        |
| Claude Code    | CLI       | 月額\$20～    |
| Gemini CLI     | CLI       | 無料～        |

### 🚀 CLIツールの利点

- 自動化しやすい - スクリプトやCIに組み込み可能
- 軽量・高速 - GUIのオーバーヘッドなし / 「VSCodeのやり方」で失敗すること無し
- 柔軟性が高い - 任意のエディタと併用可能

# VS Claude Code vs Gemini CLI

## 主な違い

| 項目     | Claude Code             | Gemini CLI           |
|--------|-------------------------|----------------------|
| 料金     | 有料（月額\$20）              | 無料でも利用可能             |
| 環境     | Linux / WindowsはWSLを使う等 | Windowsで直接動作         |
| セットアップ | 複雑（DNS設定等）              | シンプル                 |
| モデル    | Claude Opus 4/Sonnet 4  | Gemini 2.5 Pro/Flash |
| コード生成  | 非常に高品質(※主観)             | 高品質(※主観)             |
| MCP対応  | ✓                       | ✓                    |

## 💡 選び方のポイント

- 無料で試したい → Gemini CLI
- WSL設定が面倒 → Gemini CLI
- 現時点で最強のものが使いたい → Claude Code

## Step 1: インストール

### Claude Codeのインストール (Windows WSL版)

```
# WSL2のセットアップ (未インストールの場合)
```

```
wsl --install
```

```
# WSL内でNode.jsをインストール
```

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

```
# Claude Codeをインストール
```

```
npm install -g @anthropic-ai/claude-code #おそらく権限のエラーが出ます
```

 公式ドキュメント: <https://docs.anthropic.com/en/docs/claude-code>

## Claude Codeのインストール ( Windows WSL版ではこちらを個人的に推奨 )

```
# First, save a list of your existing global packages for later migration
npm list -g --depth=0 > ~/npm-global-packages.txt

# Create a directory for your global packages
mkdir -p ~/.npm-global

# Configure npm to use the new directory path
npm config set prefix ~/.npm-global

# Note: Replace ~/.bashrc with ~/.zshrc, ~/.profile, or other appropriate file for your shell
echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.bashrc

# Apply the new PATH setting
source ~/.bashrc

# Now reinstall Claude Code in the new location
npm install -g @anthropic-ai/claude-code

# Optional: Reinstall your previous global packages in the new location
# Look at ~/npm-global-packages.txt and install packages you want to keep
```

 公式ドキュメント: <https://docs.anthropic.com/en/docs/claude-code>

# ⚠ 重要！WSL環境でClaude Codeがオフラインになる問題

## 🚩 WSLでClaude Codeがすぐにofflineになる場合の解決方法

### 1. WSL設定ファイルの編集

```
# /etc/wsl.conf に以下を追加
[network]
generateResolvConf = false
```

### 2. WSLを再起動

```
wsl --shutdown
# その後、WSLを再度起動
```

### 3. DNS設定を手動で追加

```
# /etc/resolv.conf に適切なDNSサーバーを設定
echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf
```

💡 詳細はこちら → <https://note.com/ebibibi/n/n211c76198608>



# WSLとWindowsの連携

## VSCodeとWSLの連携

### Windows側のプロジェクトフォルダへのアクセス

```
# Windows側のCドライブにアクセスする場合  
cd /mnt/c
```

### パフォーマンスのために

- プロジェクトファイルはWSL側（ `~/projects/` など ）に配置推奨
- Windows側のファイルアクセスは信じられないくらい遅い

# Gemini CLIのインストール

```
# あらかじめNode.js (+npm) をインストール(インストーラーで「次へ」と進めるだけでよい)
```

```
# npmでインストール
```

```
npm install -g @google/generative-ai-cli
```

- Claude Codeと比べると超簡単！ ✨🚀
- googleアカウントは必要です

動画も参考にどうぞ！

🏠 Claude CodeをWindowsにインストールする方法！



🏠 Gemini CLI : Google最強AIが今だけ無料で使える！



## ⚙️ Step 2: 基本設定

### 📝 CLAUDE.mdの作成

- 起動時に自動的に読み込む設定ファイル
- 既存レポジトリなら `/init` を実行すれば素敵なものが自動作成される

```
# CLAUDE.mdの例
# 必ず守るべき重要な指示
- 日本語で応答すること
- 必ずMicrosoft Learn MCPを参照してから実装すること
- 初回には必ず design.md, progress.md を読み現在の状況を把握すること
- 進捗は progress.md に記載すること
```

### 📝 gemini.mdの作成

- 役割はCLAUDE.mdと一緒です。
- `/init` コマンド的なものは現時点ではなさそうです。

# 他にも山ほどTipsがあります

多すぎる & みんな試行錯誤中なので、今日は記事紹介のみとしておきます。(時間がない！)

- [速習 Claude Code](#)
- [実務で使っているClaude Codeの開発環境の紹介](#)
- [Claude Codeにコマンド一発でMCPサーバを簡単設定](#)
- [テストから始めるAgentic Coding 〜Claude Codeと共に行うTDD〜](#)  
[/ Agentic Coding starts with testing - Speaker Deck](#)

## Step 3: Microsoft Learn MCP連携

- 必須ではないですけど超お勧めです！
- MCP連携の設定場所は複数あります。下記はプロジェクトレベルの例。

### Claude Code

```
claude mcp add microsoft_learn_mcp -s project -t http https://learn.microsoft.com/
```

### Gemini CLI

- .gemini/settings.jsonに下記を記載

```
{
  "mcpServers": {
    "microsoft_learn_mcp": {
      "type": "http",
      "url": "https://learn.microsoft.com/"
    }
  }
}
```

- ただし、Gemini CLIはMCP経由ではLearnの記事の読みにいってくれない。(自力で検索しちゃう)

## Step 4: Azureへの操作はやっぱり既存ツール群

- 今さらですが…。結局devcontainerを作ってしまうのも楽だと思います。

```
# Install Azure CLI
curl -sL https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > /etc/apt/trusted.gpg.d/microsoft.gpg && \
  echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/azure-cli.list && \
  apt-get update && \
  apt-get install -y azure-cli && \
  rm -rf /var/lib/apt/lists/*

# Install PowerShell 7
wget -q "https://packages.microsoft.com/config/debian/$(lsb_release -rs)/packages-microsoft-prod.deb" -O packages-microsoft-prod.deb && \
  dpkg -i packages-microsoft-prod.deb && \
  apt-get update && \
  apt-get install -y powershell && \
  rm packages-microsoft-prod.deb && \
  rm -rf /var/lib/apt/lists/*

# Install Azure PowerShell Module
pwsh -Command "Set-PSRepository -Name 'PSGallery' -InstallationPolicy Trusted; Install-Module -Name Az -Scope AllUsers -Force"

# Install Bicep CLI
curl -Lo bicep https://github.com/Azure/bicep/releases/latest/download/bicep-linux-x64 && \
  chmod +x ./bicep && \
  mv ./bicep /usr/local/bin/bicep
```



## Step 5: AIと協力して作業する！

個人的に推奨のやり方

- まずAIと何をどのような技術要素で実装するのかを相談する。
- 決定事項を `design.md` にまとめてもらう。
- 実装計画をAIに立ててもらい `progress.md` に記載してもらう。
- AIに実装を依頼し、`progress.md` に進捗を記載してもらう。





実際にやってみましょう！

# セッション②

## Microsoft "Adaptive Cloud" Updates

スピーカー：高添 修 氏  
日本マイクロソフト株式会社

 14:25 - 14:45 ( 20分 )

# セッション③

## Azure Arc対応Kubernetes拡張機能

### "Edge RAG"

スピーカー：胡田 昌彦

日本ビジネスシステムズ株式会社

Microsoft MVP for Azure Hybrid, Windows Server

 14:45 - 15:05 ( 20分 )

## Edge RAGとは？

### Azure Arc対応Kubernetes拡張機能の最新ソリューション

#### 概要

- オンプレミスデータに生成AIを適用するRAG ( Retrieval Augmented Generation ) ソリューション
- Azure Arc対応Kubernetesクラスター上で動作する拡張機能
- プライベートデータを活用した生成AI体験をエッジで実現

#### 主な特徴

- SLM/LLM ( 小規模/大規模言語モデル ) をローカルで実行 ( CPU/GPU対応 )
- エンドツーエンドのデータ取り込みとRAGパイプライン - 全情報をオンプレミスに保持
- RBAC ( ロールベースアクセス制御 ) による安全なアクセス管理
- 100以上の言語に対応、テキスト・画像・ドキュメントをサポート ( 動画も近日対応予定 )

#### 活用シーン

- 規制やコンプライアンス要件でデータをクラウドに出せない業界
- 低遅延が求められるリアルタイム処理
- 大量のローカルデータから洞察を得たい組織

**画面を見てみましょう！**

# 環境構築で苦労したこと&Tips!

⚠ 何度も失敗するので全てスクリプト化しておくこと！

😓 展開の現実的な課題

## 失敗の頻度

- 展開には様々な場所で失敗し、何度もVM再作成からやり直した
- おそらく10回以上やり直し
- Nested Hyper-V最高！

💡 強い推奨事項

- 極力自動化することを強く推奨
- PowerShellスクリプト化

# 検証環境の最終的なスクリプト構成

```
▼ azurelocal
  ➤ 1_Deploy_NFS.ps1
  ➤ 1_PrepareAD.ps1
  ➤ 2_PrepareVM_phase1.ps1
  ➤ 3_PrepareVM_phase2.ps1
  ➤ 4_RegisterArc.ps1
  ➤ 5_DepolyLogicalNetwork.ps1
  ➤ 6_DeployAKS.ps1
  ➤ 7_ConnectAKS.ps1
  ➤ 8_CreateAKSNodePool.ps1
  ➤ 9_RegisterApp.ps1
  ➤ 10_InstallNetworkComponents.ps1
  ➤ 11_DeployEdgeRAG.ps1
  ! aks-arc-kube-config
  ⬇ EdgeRAG_Setup_Guide.md
  {} EdgeRAG-AppRegistration.json
  ≡ EdgeRAG-deployment-20250626.log
  {} EdgeRAG-deployment-result.json
  ! edgeragdemo.manifest.yaml
  ⓘ README.md
```



# Active Directory事前準備で入力する認証情報

## 個人的にわかりにくかったコマンド

```
New-HciAdObjectsPreCreation -AzureStackLCMUserCredential (Get-Credential) -AsHciOUName "OU=azurelocal,DC=dev1,DC=ebisuda,DC=net"
```

### 注意

- このADの準備段階で入力するクレデンシャルは  
azurelocalのnodeの **ローカルの管理者** となるイメージ
- 既存のドメインユーザーを入力するのではないので注意。

## AKSクラスタ作成時の注意点

### コントロールプレーンIP指定の問題

#### 問題の症状

- Edge RAGの前提条件となるAKSクラスタがコントロールプレーンIPを指定しないと展開できない状況が発生
- 本来はコントロールプレーンIPを指定しなくても展開できるはず

#### 解決方法

- 下記オプションでIPアドレスを明示することで解決

```
az aksarc create \  
# その他パラメータ省略  
--control-plane-count 1 \  
--control-plane-ip 10.1.1.205 \  

```

#### 備考

- サポートに問い合わせをしたが再現しないため未解決
- しかし、私の環境では何度でも再現した(納得いかん！🙄)

# Edge RAGの要求スペックが高すぎる問題

## 必要なVM構成

### GPUが使用可能な場合

- Standard\_NC8\_A2 × 3 ノード + Standard\_D8s\_v3 × 3 ノード

### CPUのみの場合

- Standard\_D8s\_v3 × 6 ノード

| VM サイズ          | GPU | GPU メモリ (GiB) | vCPU | メモリ (GiB) |
|-----------------|-----|---------------|------|-----------|
| Standard_NC8_A2 | 1   | 16            | 8    | 16        |

| VM サイズ          | vCPU | メモリ (GB) |
|-----------------|------|----------|
| Standard_D8s_v3 | 8    | 32       |

# 合計リソース要求量

## GPU使用時の合計スペック

- **vCPU**: 48コア ( NC8\_A2: 24 + D8s\_v3: 24 )
- **メモリ**: 120GB ( NC8\_A2: 24GB + D8s\_v3: 96GB )
- **vGPU**: 3 ( 各16GB GPU メモリ )

## CPU専用時の合計スペック

- **vCPU**: 48コア ( D8s\_v3: 8 × 6 ノード )
- **メモリ**: 192GB ( D8s\_v3: 32GB × 6 ノード )

## 現実的な課題

可用性、冗長性の観点ではよいが、個人で検証するのはきつすぎる。  
外部にLLMを配置する構成なら、1ノードだけでも動作した。

# Azure Localでのリソース管理の落とし穴

## ノードプール追加時の謎現象

### 問題の症状

- スペック上はCPU、メモリ、ストレージに余裕がある
- しかし、ノードプール追加時にVHDマウントのためのメモリが不足しているというエラーが発生

### 解決方法

### Azure Localホスト（Hyper-Vノード）の再起動

- Hyper-V的にメモリリソースが解放されない状態があった(原因不明)
- 再起動後、リソースが正常に利用可能になった

# Edge RAGの現状評価

## プレビュー版の特徴と課題

### 現状の印象

- プレビューのためまだまだ荒削り
- リソースが足りない or ギリギリだと様々な問題が発生

### 本番を見据えた設計

- 冗長構成が標準で組み込まれている
- 高可用性を前提とした構成になっている
- 障害発生時の対応も考慮された設計

### リソース要求の現実

- ホストに要求されるリソースはかなり大きい
- 個人検証には厳しいスペック要求
- しかし開発ゼロで使えるように構成されているのは良い

ありがとうございました！

# セッション①【後半】

Claude Codex×Azure, Gemini CLI×Azure

( 続き )

スピーカー：胡田 昌彦

日本ビジネスシステムズ株式会社

Microsoft MVP for Azure Hybrid, Windows Server

 15:05 - 15:15 ( 10分 )



# Q&Aセッション

 ご質問をお待ちしています！

- チャットでの質問歓迎
- 匿名での質問も可能
- どんな質問でもOK！

 15:15 - 15:25 ( 10分 )

# 次回予告



次回は8月9日（金）14:00～

毎月第2金曜日に開催！

YouTubeチャンネル登録をお忘れなく！

<https://www.youtube.com/channel/UCrf4bEl7yJnkGYo3F67gA7w>



ご参加ありがとうございました！

また来月お会いしましょう！

 本日の資料は後日公開予定

 お問い合わせは [ebibibi@gmail.com](mailto:ebibibi@gmail.com) まで

#HCCJP でツイートお願いします！