

2025-11-18 개발 세션 기록

작업: Issue #11 - Refactor VM provisioning into separate functions

작업 요약

목표

`createNetworkResources()` 함수를 분리하여 코드 모듈성과 유지보수성 향상.

완료된 작업

Branch: `feature/issue-10-refactor-provisioning`

Commits: 3개

PR: 머지 완료

Issue: #11 수동으로 닫음

Commit 1: Extract VNet creation

변경사항:

- `createVirtualNetwork()` 함수 추가 (VNet + Subnet 생성)
- `createNetworkResources()` 가 `vnet` 파라미터로 받도록 수정
- `provisionVM()` 에서 `createVirtualNetwork()` 직접 호출

코드:

```
1 async function createVirtualNetwork(networkClient, resourceGroupName, location, vmName) {
2   const vnetName = `${vmName}-vnet`;
3   const subnetName = `${vmName}-subnet`;
4   // ... VNet 생성 로직
5   return vnet;
6 }
7
```

테스트: 정상 작동 확인

Commit 2: Extract NIC creation

변경사항:

- `createNetworkResources()` → `createNetworkInterface()` 로 이름 변경
- 내부 `createPublicIP()` 호출 제거
- `vnet`, `publicIP` 를 명시적 파라미터로 받음
- `provisionVM()` 에서 모든 네트워크 함수 독립적으로 호출

개선점:

- Public IP 생성이 숨겨진 의존성에서 명시적 호출로 변경
- 각 함수의 책임이 명확해짐

코드 흐름:

```
1 const vnet = await createVirtualNetwork(...);
2 const publicIP = await createPublicIP(...);
3 const nic = await createNetworkInterface(..., vnet, publicIP);
4
```

테스트: 정상 작동 확인

Commit 3: Optimize orchestration

변경사항:

- Azure 인증과 클라이언트 초기화 분리
 - `// Azure authentication` (credential만)
 - `// Azure clients` (computeClient, resourceClient, networkClient)
- `networkClient` 초기화를 최상단으로 이동
- `vmName` 선언을 Configuration 섹션으로 이동
- TODO 주석 업데이트 (Public IP 완료 반영)

개선점:

- 코드 구조가 더 명확해짐
- 초기화 로직이 한눈에 보임

테스트: 정상 작동 확인

최종 함수 구조

Before (리팩토링 전)

```
1 provisionVM()
2   | ensureResourceGroup()
3   | createNetworkResources() ← 너무 많은 책임
4   |   | VNet 생성
5   |   | createPublicIP() 호출 (숨겨진 의존성)
6   |   | NIC 생성
7   |   | createVirtualMachine()
8
```

After (리팩토링 후)

```
1 provisionVM()
2   | ensureResourceGroup()
3   | createVirtualNetwork() ← 독립적
4   | createPublicIP() ← 독립적
5   | createNetworkInterface() ← vnet, publicIP 명시적 의존성
6   | createVirtualMachine() ← nic 명시적 의존성
7
```

주요 학습 포인트

1. Single Responsibility Principle

- 각 함수는 하나의 책임만 가져야 함
- `createVirtualNetwork()` : VNet만
- `createPublicIP()` : Public IP만
- `createNetworkInterface()` : NIC만

2. Explicit Dependencies

- 함수 내부에서 다른 함수를 호출하지 않음
- 필요한 모든 것을 파라미터로 받음
- 의존성이 명확하게 보임

3. Orchestration Layer

- `provisionVM()` 이 오케스트레이션 담당
- 전체 흐름이 한 곳에서 명확하게 보임

4. Issue 번호 확인의 중요성

- PR에서 `Closes #10` 이라고 했는데 실제로는 #11
 - 자동 닫힘 실패, 수동으로 처리
 - 다음부터는 Issue 번호 확인 필수
-

Git 워크플로우

```
1 # 1. 브랜치 생성
2 git checkout main
3 git pull origin main
4 git checkout -b feature/issue-10-refactor-provisioning
5
6 # 2. 작업 (3 commits)
7 git add lib/vmManager.js
8 git commit -m "...
9
10 # 3. Push
11 git push origin feature/issue-10-refactor-provisioning
12
13 # 4. PR 생성 및 머지
14
15 # 5. 정리
16 git checkout main
17 git pull origin main
18 git branch -d feature/issue-10-refactor-provisioning
19
```

Phase 3 진행 상황

완료:

- ☒ Issue #5: Public IP allocation and assignment
- ☒ Issue #11: Refactor VM provisioning (원래 #10)

남은 작업:

- 🕒 Issue #6: Implement cloud-init for noVNC installation
- 🕒 Issue #9: Test noVNC web access and VM functionality

다음 세션 준비

시작 명령어

```
1 cd C:\Users\blitz\Desktop\GBL-Square\unity-ci-license-activator-azure
2 git status
3 git pull origin main
4 az account show
5
```

다음 작업: Issue #6 (cloud-init)

- noVNC 설치 스크립트 작성
- cloud-init configuration 추가
- `createVirtualMachine()` 에 `customData` 추가

작업 시간: ~2시간

코드 변경: lib/vmManager.js

테스트: 각 커밋마다 로컬 테스트 완료

Status: Issue #11 완료 ☒