

2025-11-10 개발 세션 기록

📅 Today's Summary

Phase 1 복습 완료 및 Phase 2 프로젝트 관리 구조 확립

✅ Completed Tasks

1. Phase 1 아키텍처 복습

Azure Functions 레이어 구조:

- Node.js Runtime (기반)
- azure-functions-core-tools (CLI 도구)
- Azure Functions Runtime (프레임워크)
- Application Code (개발자 작성)

각 파일 역할:

- `host.json` : Functions Runtime 전역 설정
- `function.json` : 트리거/바인딩 설정
- `index.js` : HTTP 요청 처리 및 HTML 응답
- `vmManager.js` : VM 생성 비즈니스 로직

ES Modules vs CommonJS:

- ES Modules: `import/export` (최신 표준)
- CommonJS: `require/module.exports` (전통 방식)

2. 실행 환경 및 인증 전략

DefaultAzureCredential 동작:

- 로컬: Azure CLI 인증
- Production: Managed Identity
- 같은 코드로 두 환경 지원

```
1 const credential = new DefaultAzureCredential();  
2 const client = new ComputeManagementClient(credential, subscriptionId);
```

인증 구분:

- Phase 2: Azure 인증 (VM 생성)
- Phase 5: GitHub 인증 (Webhook)

3. 프로젝트 관리 구조

생성 완료:

- Milestone: Phase 2: Azure Authentication & VM Provisioning
- Issue #2: Setup Azure authentication for local development
- Project: Unity CI License Activator - Development

Issue #2:

- Description: Azure 로컬 개발 환경 구성
 - Tasks: CLI 설치, az login, local.settings.json 등
 - Acceptance Criteria: 인증 성공, 에러 없음
-

Key Learnings

DefaultAzureCredential 자동 감지:

1. 환경 변수
2. Managed Identity (Azure)
3. Azure CLI (로컬)
4. Visual Studio
5. IntelliJ

로컬에서 Azure 리소스 생성: 로컬 코드 → 인터넷 → Azure API → 클라우드에 VM 생성

Project Structure

```
1 unity-ci-license-activator-azure/
2 |─ function-setup/
3 | |─ function.json
4 | |─ index.js
5 |─ lib/
6 | |─ vmManager.js
7 |─ host.json
8 |─ package.json
```

Next Steps

Issue #2 시작:

```
1 az --version
2 az login
3 az account show
```