2025-11-11 개발 세션 기록

📋 Today's Development Session

Date: November 12, 2025
Session Focus: Phase 2 - Azure Authentication Setup

✅ Completed Tasks

1. Azure CLI Setup and Authentication

Installation:

- Installed Azure CLI (version 2.79.0)
- Verified installation via PowerShell
- Successfully authenticated with `az login`

Account Configuration:

- Selected "Azure for Students" subscription
- Subscription ID: `2faf0e7b-616e-44d6-aea9-7b54f0664d84`
- Tenant: Conestoga College

2. Environment Configuration

Created `local.settings.json`:

```
1  {
2    "IsEncrypted": false,
3    "Values": {
4      "AzureWebJobsStorage": "",
5      "FUNCTIONS_WORKER_RUNTIME": "node",
6      "AZURE_SUBSCRIPTION_ID": "2faf0e7b-616e-44d6-aea9-7b54f0664d84",
7      "RESOURCE_GROUP_NAME": "unity-ci-vms",
8      "AZURE_LOCATION": "eastus"
9    }
10 }
```

Key Decisions:

- All configuration values managed in `local.settings.json` (no hardcoded defaults)
- File is gitignored for security
- Environment variables provide flexibility for different deployment environments

3. Code Implementation

Modified `function-setup/index.js`:

- Added subscription ID validation from environment variables
- Returns 500 error if `AZURE_SUBSCRIPTION_ID` not configured
- Passes `subscriptionId` to `createVM()` function
- Improved error handling

Modified `lib/vmManager.js`:

- Added `subscriptionId` parameter to `createVM()` function
- Initialized `DefaultAzureCredential` for Azure authentication
- Initialized `ComputeManagementClient` with credential and subscription
- Configured resource group and location from environment variables
- Added authentication test logging

4. Testing and Verification

Local Test Results:

```
1  [2025-11-12T04:04:00.597Z] Creating VM: vm-1762920240594
2  [2025-11-12T04:04:00.599Z] Using subscription: 2faf0e7b-616e-44d6-aea9-
   7b54f0664d84
3  [2025-11-12T04:04:00.611Z] Azure authentication successful
4  [2025-11-12T04:04:00.613Z] Compute client initialized
5  [2025-11-12T04:04:00.635Z] Setup function called
6  [2025-11-12T04:04:00.637Z] VM creation started
```

Verified:

- ✅ Azure CLI authentication working
- ✅ Environment variables loaded correctly
- ✅ `DefaultAzureCredential` successfully obtained token
- ✅ `ComputeManagementClient` initialized without errors
- ✅ No authentication or configuration errors

5. Git Commits

Commit 1 (ca8ae84):

```
1  feat: add Azure subscription ID environment variable support
2
3  - Read AZURE_SUBSCRIPTION_ID from environment variables
4  - Add validation and error handling for missing subscription ID
5  - Pass subscriptionId to createVM() function
6  - Return 500 error if configuration is missing
```

Commit 2 (241e59f):

```
1  feat: implement Azure authentication in VM manager
2
3  - Add DefaultAzureCredential for authentication
4  - Initialize ComputeManagementClient with subscription ID
5  - Add subscriptionId parameter to createVM function
6  - Configure resource group and location from environment variables
7  - Add authentication test logging
```

6. Project Management

Issue #2 Completed:

- Title: Setup Azure authentication for local development
- Status: Closed
- All tasks completed successfully

New Issues Created:

- **Issue #3:** Implement VM creation with Azure Compute API
  - Milestone: Phase 2
  - Focus: API implementation and configuration
- **Issue #4:** Test and verify VM creation in Azure
  - Milestone: Phase 2
  - Focus: End-to-end testing and validation
  - Dependency: Issue #3

🎯 Key Technical Insights

DefaultAzureCredential Authentication Chain

Current (Local Development):

```
1  const credential = new DefaultAzureCredential();
2  // → Searches: Environment Variables → Managed Identity → Azure CLI ✓
3  // → Uses: Azure CLI token from `az login`
```

Future (Production):

```
1  const credential = new DefaultAzureCredential();
2  // → Searches: Environment Variables → Managed Identity ✓
3  // → Uses: Managed Identity (auto-configured in Azure Functions)
```

Key Understanding:

- Same code works in both environments
- No code changes needed for deployment
- Authentication automatically adapts to context

What We Actually Verified

Confirmed:

- Azure CLI authentication token exists and is valid
- `DefaultAzureCredential` successfully retrieved token
- `ComputeManagementClient` object created without errors
- SDK initialization successful

Not Yet Tested:

- Actual Azure API calls
- VM creation permissions
- Resource provisioning

Analogy:

- Created the tools and verified they're ready
- Haven't actually built anything yet
- Like confirming you have a hammer before building furniture

Configuration Strategy

Centralized in `local.settings.json`:

- Single source of truth for environment configuration

- No default values in code (explicit configuration required)

- Clear separation of code and configuration

- Easy to modify for different environments

📁 Current Project State

File Structure

```
 1  unity-ci-license-activator-azure/
 2  ├── function-setup/
 3  │   ├── function.json
 4  │   └── index.js                [Modified]
 5  ├── lib/
 6  │   └── vmManager.js             [Modified]
 7  ├── local.settings.json          [Created, gitignored]
 8  ├── host.json
 9  ├── package.json
10  └── .gitignore
```

Git Status

- Branch: `main`

- Commits: 2 new commits today

- Uncommitted changes: None

- Remote: Synced

Issue Tracker

- **Phase 2 Milestone:** 1/3 issues completed

  - ✅ Issue #2: Setup Azure authentication

  - 🔄 Issue #3: Implement VM creation

  - ⏳ Issue #4: Test VM creation

🚀 Next Session Plan

Immediate Next Steps (Issue #3)

1. **Resource Group Management:**

   - Check if resource group exists

   - Create if not exists

   - Handle resource group errors

2. **VM Configuration:**

   - Define VM parameters (size, image, OS disk)

   - Configure networking (VNet, subnet, NIC)

   - Set up basic security

3. **API Implementation:**

   - Call `virtualMachines.beginCreateOrUpdate()`

   - Handle long-running operation (LRO)

   - Add progress logging

4. **Error Handling:**
  - Permission errors
  - Quota exceeded
  - Network configuration failures

Session Start Commands

```
1  cd C:\Users\blitz\Desktop\GBL-Square\unity-ci-license-activator-azure
2  git status
3  git pull
4  npx func start
```

📝 Notes and Observations

Decision: Environment-Only Configuration

Question Raised: Should we use `|| 'default-value'` pattern or require all values in environment?

Decision: Environment variables only, no defaults in code

Rationale:

- Clearer intent (values must be explicitly configured)
- Prevents silent failures with wrong defaults
- Easier to spot missing configuration
- Better for multiple deployment environments

Understanding: SDK Initialization vs API Calls

Clarification: Successfully creating `ComputeManagementClient` doesn't mean we've made any API calls yet.

What Happened:

- Created credential object (obtained token)
- Created client object (prepared to make calls)
- No actual HTTP requests to Azure API

Verification Pending:

- Actual permission to create VMs
- API call success/failure
- Resource creation

IDE Warning: AZURE_SUBSCRIPTION_ID

Issue: WebStorm shows "unresolved variable" warning

Cause: IDE doesn't read `local.settings.json`

Resolution: Ignore warning (code works correctly at runtime)

🔧 Technical Environment

Tools & Versions

- **OS:** Windows 10/11
- **Node.js:** v22.20.0
- **Azure CLI:** 2.79.0
- **IDE:** WebStorm/IntelliJ IDEA Ultimate
- **Shell:** PowerShell

Azure Resources

Current: None (mock data only)

After Issue #3:

- Resource Group: `unity-ci-vms`
- VM: Test instance (Standard_B1s)
- Location: East US

Session Duration: ~2 hours
Lines of Code Changed: ~50
Issues Resolved: 1
Issues Created: 2