2025-11-14 개발 세션 기록
📋 Today's Development Session

Date: November 15, 2025

Session Focus: Phase 2 Completion & Phase 3 Planning

✅ Completed Tasks

1. VM Creation Implementation (Issue #3)

Package Installation:

```
npm install @azure/arm-network @azure/arm-resources
```

Added dependencies:

- `@azure/arm-network@33.2.0` - Virtual Network and NIC management
- `@azure/arm-resources@5.2.0` - Resource Group management

Code Implementation (3 commits):

Commit 1: Resource Group Creation

```
// Added ensureResourceGroup() helper function
async function ensureResourceGroup(resourceClient, resourceGroupName, location)
```

- Checks if resource group exists
- Creates if not found (404 error)
- Reuses existing if already present
- Comprehensive error handling

Commit 2: Network Resources

```
// Added createNetworkResources() helper function
async function createNetworkResources(networkClient, resourceGroupName, location, vmName)
```

- Creates Virtual Network (10.0.0.0/16)
- Creates Subnet (10.0.0.0/24)
- Creates Network Interface
- Attaches NIC to subnet
- Returns NIC with private IP

Commit 3: VM Creation

```
// Added createVirtualMachine() helper function
async function createVirtualMachine(computeClient, resourceGroupName, location, vmName, nic)
```

- Ubuntu Server 22.04 LTS image
- VM Size: Standard_B1s (1 vCPU, 1 GiB RAM)
- Disk: Standard_LRS (HDD)
- Admin credentials: azureuser / Azure123456!
- Network profile with NIC attachment

Function Refactoring:

- Renamed `createVM()` → `provisionVM()`
- Better reflects full provisioning workflow
- Updated imports in `function-setup/index.js`

2. Git Workflow & Branch Management

Branch Strategy:

```
main
 └─ feature/issue-3-vm-creation
     ├─ Commit 1: Resource group helper
     ├─ Commit 2: Network resources
     └─ Commit 3: VM creation
```

Pull Request #7:

- Title: "Implement VM creation with Azure Compute API"
- Comprehensive description with technical details
- Linked to Issue #3 via `Closes #3`
- Merged to main successfully

Workflow Learning:

- Separate branches per issue for clean history
- Multiple logical commits within feature branch
- Each commit represents working state
- PR descriptions document changes thoroughly

3. End-to-End Testing (Issue #4)

Test Execution:

```
npx func start
# Navigate to http://localhost:7071/api/function-setup
```

Console Output:

```
[2025-11-15T03:03:28.195Z] Creating VM: vm-1763175808192
[2025-11-15T03:03:28.197Z] Using subscription: 2faf0e7b-616e-44d6-aea9-7b54f0664d84
[2025-11-15T03:03:28.204Z] Checking resource group: unity-ci-vms
[2025-11-15T03:03:30.897Z] Resource group unity-ci-vms already exists
[2025-11-15T03:03:30.899Z] Resource group ready
[2025-11-15T03:03:30.901Z] Creating virtual network...
[2025-11-15T03:03:53.615Z] Virtual network created (23 seconds)
[2025-11-15T03:03:53.616Z] Creating network interface...
[2025-11-15T03:03:53.988Z] Network interface created (0.4 seconds)
[2025-11-15T03:03:53.989Z] Network resources ready
[2025-11-15T03:03:53.991Z] Creating virtual machine...
[2025-11-15T03:04:09.641Z] Virtual machine created: vm-1763175808192 (16 seconds)
[2025-11-15T03:04:09.643Z] VM creation completed successfully
```

Total Provisioning Time: ~41 seconds

Azure Resources Created:

- Resource Group: `unity-ci-vms` (East US)
- Virtual Machine: `vm-1763175808192`
- Virtual Network: `vm-1763175808192-vnet`
- Subnet: `vm-1763175808192-subnet`
- Network Interface: `vm-1763175808192-nic`
- OS Disk: `vm-1763175808192_disk1_ad46a8f7cf614a43b81152c505d74947`

Verification:

- ✅ All resources visible in Azure Portal
- ✅ VM status: Running
- ✅ Private IP assigned: 10.0.0.4
- ✅ No authentication errors
- ✅ No quota issues

4. Test Documentation

Created: `docs/testing/phase-2-vm-creation.md`

Documentation Structure:

```
docs/
└── testing/
    └── phase-2-vm-creation.md
```

Contents:

- Test information and environment details
- Execution steps and commands
- Detailed test results with specifications
- Performance metrics table
- Console output logs
- Azure Portal screenshots (3)
- Issues and observations
- Acceptance criteria verification
- Resource cleanup documentation

Key Metrics Documented:

Resource	Creation Time
Resource Group	< 1s (existed)
Virtual Network	23s
Network Interface	0.4s
Virtual Machine	16s
Total	**~41s**

5. Resource Cleanup

Method: Azure Portal - Delete Resource Group

Deleted Resources:

- All VMs created during testing
- All network resources (VNets, NICs)
- All managed disks
- Resource Group: `unity-ci-vms`

Cost Savings:

- Prevented ongoing charges (~$0.24/day per VM)
- Clean environment for Phase 3

6. Pull Request Management

PR #8:

- Title: "Test and verify VM creation in Azure"
- Branch: `feature/issue-4-vm-testing`
- Comprehensive test results documented
- Screenshots included in description
- Linked to Issue #4 via `Closes #4`
- Merged successfully

7. Phase 3 Planning

Milestone Created:

- Title: "Phase 3: Public IP & noVNC Setup"
- Due date: November 22, 2025
- Description with clear goals

Issues Created:

Issue #5: Allocate and assign Public IP to VM

- Tasks: Public IP resource creation, NIC attachment, metadata updates
- Acceptance criteria defined
- Technical notes with API references

Issue #6: Implement cloud-init for noVNC installation

- Tasks: Script creation, Desktop installation, noVNC setup
- Detailed technical requirements
- cloud-init configuration approach

Issue #9: Test noVNC web access and VM functionality

- End-to-end testing focus
- Browser access verification
- Dependencies on #5 and #6

GitHub Projects Configuration:

- All issues added to project board
- Status: Todo
- Estimates added
- Priority and Size fields explored

🎯 Key Technical Insights

Azure SDK Architecture

Helper Functions Pattern:

```
// Modular helper functions for each responsibility
async function ensureResourceGroup()      // Resource management
async function createNetworkResources()   // Network provisioning
async function createVirtualMachine()     // VM creation
export async function provisionVM()       // Orchestrator
```

Benefits:

- Clear separation of concerns
- Easier testing and debugging
- Reusable components
- Better error isolation

Long-Running Operations (LRO)

Pattern Used:

```
await networkClient.virtualNetworks.beginCreateOrUpdateAndWait(...)
await computeClient.virtualMachines.beginCreateOrUpdateAndWait(...)
```

Understanding:

- Azure operations are asynchronous
- `beginCreateOrUpdateAndWait()` blocks until completion
- Alternative: `beginCreateOrUpdate()` returns immediately (polling required)

- Sequential execution ensures dependencies met

Network Configuration

IP Address Allocation:

```
VNet: 10.0.0.0/16 (65,536 addresses)
  └─ Subnet: 10.0.0.0/24 (256 addresses)
       └─ VM NIC: 10.0.0.4 (DHCP assigned)
```

Azure Reserved IPs:

- 10.0.0.0: Network address
- 10.0.0.1: Default gateway
- 10.0.0.2: Azure DNS
- 10.0.0.3: Reserved for future use
- 10.0.0.4: First usable address (our VM)

Hard-Coded vs Environment Variables

Decision Matrix:

Environment Variables (local.settings.json):

- ✅ AZURE_SUBSCRIPTION_ID (sensitive)
- ✅ RESOURCE_GROUP_NAME (user-specific)
- ✅ AZURE_LOCATION (deployment-specific)

Hard-Coded (vmManager.js):

- ✅ Network CIDR blocks (10.0.0.0/16, /24)
- ✅ VM size (Standard_B1s)
- ✅ OS image (Ubuntu 22.04)
- ✅ IP config names (ipconfig1)

Rationale:

- Network configs are standard and unlikely to change
- VM specs can be parameterized later if needed
- Reduces configuration complexity
- Clearer code intent

Async Execution Warning

Warning Observed:

```
Warning: Unexpected call to 'log' on the context object after function execution has completed.
```

Cause:

```
// index.js
provisionVM(vmId, subscriptionId)  // No await
    .then(vm => {
        context.log('VM creation started:', vm);  // Executes after response sent
    })
```

Impact:

- None on functionality
- VM still creates successfully
- HTTP response already sent to client

Future Resolution:

- Consider async/await pattern in index.js
- Or callback-based completion notification
- Not blocking for current phase

📁 Current Project State

File Structure

```
unity-ci-license-activator-azure/
├── function-setup/
│   ├── function.json
│   └── index.js                    [Modified - provisionVM import]
├── lib/
│   └── vmManager.js                [Complete rewrite]
├── docs/
│   └── testing/
│       └── phase-2-vm-creation.md  [New]
├── .github/
│   └── ISSUE_TEMPLATE/
│       └── phase-implementation.md
├── local.settings.json             [gitignored]
├── host.json
├── package.json                    [Updated dependencies]
├── package-lock.json               [Updated]
└── .gitignore
```

Git Status

- **Branch:** main
- **Latest Commit:** PR #8 merge
- **Uncommitted Changes:** None
- **Remote:** Fully synced

Issue Tracker

Phase 2 Milestone: 100% Complete ✅

- ✅ Issue #2: Azure authentication
- ✅ Issue #3: VM creation implementation
- ✅ Issue #4: Testing and verification

Phase 3 Milestone: 0% Complete

- ⏳ Issue #5: Public IP allocation
- ⏳ Issue #6: cloud-init/noVNC
- ⏳ Issue #9: End-to-end testing

🚀 Next Session Plan

Issue #5: Public IP Allocation

Session Start Commands:

```
cd C:\Users\blitz\Desktop\GBL-Square\unity-ci-license-activator-azure
git checkout main
git pull
git checkout -b feature/issue-5-public-ip
```

Implementation Steps:

1. **Create Public IP Helper Function:**

```
async function createPublicIP(networkClient, resourceGroupName, location, vmName) {
    const publicIPName = `${vmName}-ip`;
    const publicIPParams = {
        location: location,
        publicIPAllocationMethod: 'Static',
        sku: { name: 'Basic' }
    };
    return await networkClient.publicIPAddresses.beginCreateOrUpdateAndWait(
        resourceGroupName,
        publicIPName,
        publicIPParams
    );
}
```

2. **Update createNetworkResources():**

- Call createPublicIP() before creating NIC
- Attach Public IP to NIC's ipConfigurations
- Return both NIC and Public IP

3. **Update provisionVM() Return:**

```
return {
    vmId: vmId,
    status: 'succeeded',
    vmName: vm.name,
    resourceGroup: resourceGroupName,
    location: location,
    privateIP: nic.ipConfigurations[0].privateIPAddress,
    publicIP: publicIP.ipAddress  // New field
};
```

4. **Testing:**

- Verify Public IP creation in Azure Portal
- Test external connectivity (ping test)
- Document provisioning time impact

Acceptance Criteria:

- [ ] Public IP resource created
- [ ] IP attached to NIC
- [ ] Static allocation confirmed
- [ ] Public IP returned in metadata
- [ ] Portal verification successful

📝 Notes and Observations

Learning: Git Branch Workflow

Pattern Established:

```
Feature Branch → Multiple Commits → PR → Review → Merge → Delete Branch
```

Benefits Experienced:

- Clean main branch history
- Isolated feature development
- Easy rollback if needed
- Clear PR descriptions for documentation

Decision: Function Naming

Question: Why rename `createVM()` to `provisionVM()`?

Answer:

- `createVM` implies only VM creation
- Function actually does: Resource Group + Network + VM
- `provisionVM` better describes full workflow
- Industry term "provisioning" = complete resource setup

Understanding: GitHub Issue Hierarchy

Clarification on Sub-issues:

Not Suitable for Our Use:

```
Issue #5 (Public IP)
├─ Sub: Issue #6 (cloud-init) ❌ Wrong!
└─ Sub: Issue #9 (Testing) ❌ Wrong!
```

Reason: #6 and #9 are not subtasks of #5

When to Use Sub-issues:

```
Issue: Implement Authentication System (Epic)
├─ Sub: Build login API
├─ Sub: Add OAuth
├─ Sub: Password reset
└─ Sub: 2FA
```

Our Structure (Correct):

```
Milestone: Phase 3 (Epic level)
├─ Issue #5 (Feature level)
├─ Issue #6 (Feature level)
└─ Issue #9 (Feature level)
```

GitHub Projects Fields

Estimate: Story points or time units

- Relative complexity measure
- Team-defined scale
- Example: 1=simple, 7=complex

Priority: Task urgency

- P0: Critical/Blocker
- P1: High
- P2: Medium
- P3: Low

Size: Work magnitude

- XS: <1 hour
- S: 2-4 hours
- M: 1 day
- L: 2-3 days
- XL: 1+ week

Our Configuration:

- Issue #5: Priority P1, Size M
- Issue #6: Priority P1, Size L
- Issue #9: Priority P2, Size S

Cost Management

Azure Resources Cost (If Not Deleted):

- Standard_B1s VM: ~$0.01/hour = ~$0.24/day = ~$7.20/month
- Standard_LRS Disk: ~$0.20/month
- Public IP (Static): ~$3/month

Student Subscription:

- $100 credit
- Free tier available for some services
- Important to delete test resources

Best Practice: Delete immediately after testing

🔧 Technical Environment

Tools & Versions:

- OS: Windows 10/11
- Node.js: v22.20.0

- npm: 10.9.3
- Azure CLI: 2.79.0
- IDE: WebStorm
- Shell: PowerShell

Azure Configuration:

- Subscription: Azure for Students
- Subscription ID: 2faf0e7b-616e-44d6-aea9-7b54f0664d84
- Default Location: East US
- Resource Group: unity-ci-vms (deleted after testing)

Packages Added:

```
"dependencies": {
  "@azure/identity": "^4.13.0",
  "@azure/arm-compute": "^23.1.0",
  "@azure/arm-network": "^33.2.0",    // New
  "@azure/arm-resources": "^5.2.0"    // New
}
```

📊 Session Statistics

Duration: ~3 hours
Commits: 4 (3 in Issue #3, 1 in Issue #4)
Pull Requests: 2 (#7, #8)
Lines of Code Added: ~200
Issues Resolved: 2 (#3, #4)
Issues Created: 3 (#5, #6, #9)
Milestones Completed: 1 (Phase 2)
Milestones Created: 1 (Phase 3)
Documentation Files: 1 (phase-2-vm-creation.md)
Azure Resources Created: 6
Azure Resources Deleted: 6

🎓 Key Takeaways

1. **Modular Code Structure:** Helper functions make complex workflows manageable
2. **Git Workflow:** Feature branches + PRs = clean history
3. **Documentation:** Comprehensive test docs valuable for portfolio
4. **Azure SDK:** beginCreateOrUpdateAndWait pattern for sequential dependencies
5. **Cost Management:** Always clean up test resources
6. **Issue Management:** Milestones group features, sub-issues for task breakdown
7. **Environment Config:** Separate sensitive/variable data from constants

---

Phase 2: Complete! 🎉
Ready for Phase 3: Public IP & noVNC Setup