MyCareCRM Deployment Workflow - Detailed Process

Current Manual Process (To Be Automated)

This document details the exact steps currently followed for MyCareCRM deployments and maps them to automation tasks.

Step 1: Portal Configuration & Payload Preparation

Current Manual Process

1. Add client to portal
2. Send payload from portal
3. Access JUMP Host desktop
4. Use `deploy_dataset 1.json` file
5. Add `.sh` runners to portal for easy installation

Automation Requirements

- □ Integrate with portal API (if available)
- □ Automate payload generation from portal data
- □ Store payload in standardized format
- □ Trigger AWX deployment from portal

AWX Integration

Input: Client data from portal

Output: Standardized JSON payload for AWX

Payload Structure (deploy_dataset 1.json):

JSON

```
{
  "company_code": "SHORTNAME",
  "environment": "prod",
  "hyperv_host": "hyperv_prod1",
  "vlan_id": "XXX",
  "vm_app_name": "MYCC-SHORTNAME-APP",
  "vm_db_name": "MYCC-SHORTNAME-DB",
  "vm_api_name": "MYCC-SHORTNAME-API",
  "vm_app_ip": "10.12.XXX.10",
  "vm_db_ip": "10.12.XXX.11",
  "vm_api_ip": "10.12.XXX.12",
  "vm_app_cpu": 4,
  "vm_app_memory": 8192,
  "vm_db_cpu": 8,
  "vm_db_memory": 16384,
  "vm_api_cpu": 4,
  "vm_api_memory": 8192,
  "vm_app_disk": 100,
  "vm_db_disk": 200,
  "vm_api_disk": 100,
  "network_gateway": "10.12.XXX.1",
  "network_dns": "10.12.0.1",
  "network_subnet": "255.255.255.0",
  "domain_name": "shortname.mycarecrm.com.au",
  "api_domain_name": "api-shortname.mycarecrm.com.au"
}
```

Automation Task:

- Create AWX job template variable prompt
- Accept JSON payload from portal webhook
- Validate payload structure
- Launch Task 15 deployment

Step 2: DNS Configuration (Synergy Wholesale)

Current Manual Process

1. Navigate to https://manage.synergywholesale.com
2. Add A record manually
3. Configure DNS for `shortname.mycarecrm.com.au`
4. Configure DNS for `api-shortname.mycarecrm.com.au`

DNS Records Required

Plain Text

```
# A Records (pointing to Sophos public IP)
shortname.mycarecrm.com.au.        IN A    [Sophos_Public_IP]
api-shortname.mycarecrm.com.au.    IN A    [Sophos_Public_IP]

# Optional CNAME
www-shortname.mycarecrm.com.au.    IN CNAME shortname.mycarecrm.com.au.
```

Automation Requirements

- □ Research Synergy Wholesale API documentation
- □ Create API credentials for automation
- □ Develop Ansible module/playbook for DNS creation
- □ Integrate into Task 15 workflow
- □ Add DNS cleanup to decommission playbook

Synergy Wholesale API Integration

API Documentation: https://www.synergywholesale.com/support/api-documentation/

Ansible Playbook (Step 2A - DNS Creation):

YAML

```yaml
---
- name: Step 2A - Create DNS Records in Synergy Wholesale
  hosts: localhost
  gather_facts: no

  tasks:
    - name: Create A record for APP domain
      uri:
        url: "https://api.synergywholesale.com/dns/add-record"
        method: POST
        headers:
          Authorization: "Bearer {{ synergy_api_key }}"
        body:
          domain: "mycarecrm.com.au"
          name: "{{ company_code }}"
          type: "A"
          content: "{{ sophos_public_ip }}"
          ttl: 3600
        body_format: json
      register: dns_app_result
    - name: Create A record for API domain
      uri:
        url: "https://api.synergywholesale.com/dns/add-record"
        method: POST
        headers:
          Authorization: "Bearer {{ synergy_api_key }}"
        body:
          domain: "mycarecrm.com.au"
          name: "api-{{ company_code }}"
          type: "A"
          content: "{{ sophos_public_ip }}"
          ttl: 3600
        body_format: json
      register: dns_api_result

    - name: Wait for DNS propagation
      wait_for:
        timeout: 30

    - name: Verify DNS resolution
      command: "dig +short {{ company_code }}.mycarecrm.com.au"
      register: dns_check
      retries: 5
      delay: 10
      until: dns_check.stdout == sophos_public_ip
```

AWX Credential Type: Synergy Wholesale API

YAML

```
Input Configuration:
  fields:
    - id: synergy_api_key
      type: string
      label: Synergy Wholesale API Key
      secret: true
    - id: sophos_public_ip
      type: string
      label: Sophos Public IP Address

Injector Configuration:
  env:
    SYNERGY_API_KEY: "{{ synergy_api_key }}"
    SOPHOS_PUBLIC_IP: "{{ sophos_public_ip }}"
```

Step 3: UniFi VLAN Assignment

Current Manual Process

1. Navigate to https://unifi.ui.com/consoles
2. Manually assign VLAN to client
3. Configure VLAN settings

Automation Requirements

- □ Determine if VLAN is pre-created or created on-demand
- □ Get UniFi Controller API credentials
- □ Develop Ansible playbook for VLAN assignment
- □ Integrate into Task 15 workflow

UniFi VLAN Assignment

Ansible Playbook (Step 3 - UniFi VLAN):

YAML

```yaml
---
- name: Step 3 - Assign UniFi VLAN
  hosts: localhost
  gather_facts: no

  vars:
    unifi_controller: "{{ lookup('env', 'UNIFI_CONTROLLER_URL') }}"
    unifi_username: "{{ lookup('env', 'UNIFI_USERNAME') }}"
    unifi_password: "{{ lookup('env', 'UNIFI_PASSWORD') }}"

  tasks:
    - name: Login to UniFi Controller
      uri:
        url: "{{ unifi_controller }}/api/login"
        method: POST
        body:
          username: "{{ unifi_username }}"
          password: "{{ unifi_password }}"
        body_format: json
        validate_certs: no
      register: unifi_login

    - name: Create VLAN network for client
```

```
      uri:
        url: "{{ unifi_controller }}/api/s/default/rest/networkconf"
        method: POST
        headers:
          Cookie: "{{ unifi_login.set_cookie }}"
        body:
          name: "MYCC-{{ company_code }}-vlan"
          purpose: "corporate"
          vlan_enabled: true
          vlan: "{{ vlan_id }}"
          dhcpd_enabled: false
          ipsubnet: "10.12.{{ vlan_id }}.0/24"
          domain_name: "{{ company_code }}.local"
        body_format: json
        validate_certs: no
      register: vlan_result

    - name: Display VLAN creation result
      debug:
        msg: "VLAN {{ vlan_id }} created for {{ company_code }}"
```

Step 4: Sophos Firewall Configuration

Current Manual Process

4.1: Add Hosts and Services

Add the following hosts:

- **MYCC-SHORTNAME-APP** - `10.12.vlan.10`
- **MYCC-SHORTNAME-DB** - `10.12.vlan.11`
- **MYCC-SHORTNAME-API** - `10.12.vlan.12`

4.2: Add VM to Web Server (WAF)

Configure web servers for WAF:

- **MYCC-SHORTNAME-APP** - Host "MYCC-SHORTNAME-APP" port 80
- **MYCC-SHORTNAME-API** - Host "MYCC-SHORTNAME-API" port 8000

4.3: Rules and Policies

- Clone WAF Rule #149 → Link to `shortname.mycarecrm.com.au` → ENABLE

- Clone WAF Rule #150 → Link to `api-shortname.mycarecrm.com.au` → ENABLE

Automation Requirements

- □ Get Sophos XG Firewall API credentials
- □ Research Sophos API for host creation
- □ Research Sophos API for WAF configuration
- □ Develop Ansible playbook for Sophos configuration
- □ Integrate into Task 15 workflow

Sophos Firewall Automation

Ansible Playbook (Step 4 - Sophos Configuration):

YAML

```yaml
---
- name: Step 4 - Configure Sophos Firewall
  hosts: localhost
  gather_facts: no

  vars:
    sophos_api_url: "{{ lookup('env', 'SOPHOS_API_URL') }}"
    sophos_api_token: "{{ lookup('env', 'SOPHOS_API_TOKEN') }}"

  tasks:
    # 4.1 - Create Host Objects
    - name: Create APP host object
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController?reqxml=<Request>
<Login><Username>{{ sophos_username }}</Username><Password>{{
sophos_password }}</Password></Login><Set operation='add'><IPHost><Name>MYCC-
{{ company_code | upper }}-APP</Name><IPFamily>IPv4</IPFamily>
<HostType>IP</HostType><IPAddress>10.12.{{ vlan_id }}.10</IPAddress></IPHost>
</Set></Request>"
        method: POST
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: app_host_result

    - name: Create DB host object
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController?reqxml=<Request>
<Login><Username>{{ sophos_username }}</Username><Password>{{
```

```
sophos_password }}</Password></Login><Set operation='add'><IPHost><Name>MYCC-
{{ company_code | upper }}-DB</Name><IPFamily>IPv4</IPFamily>
<HostType>IP</HostType><IPAddress>10.12.{{ vlan_id }}.11</IPAddress></IPHost>
</Set></Request>"
        method: POST
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: db_host_result

    - name: Create API host object
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController?reqxml=<Request>
<Login><Username>{{ sophos_username }}</Username><Password>{{
sophos_password }}</Password></Login><Set operation='add'><IPHost><Name>MYCC-
{{ company_code | upper }}-API</Name><IPFamily>IPv4</IPFamily>
<HostType>IP</HostType><IPAddress>10.12.{{ vlan_id }}.12</IPAddress></IPHost>
</Set></Request>"
        method: POST
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: api_host_result

    # 4.2 - Create Web Server Objects for WAF
    - name: Create APP web server object
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="add">
              <WebServer>
                <Name>MYCC-{{ company_code | upper }}-APP</Name>
                <Host>MYCC-{{ company_code | upper }}-APP</Host>
                <Port>80</Port>
                <Protocol>HTTP</Protocol>
              </WebServer>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: app_webserver_result
    - name: Create API web server object
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="add">
              <WebServer>
                <Name>MYCC-{{ company_code | upper }}-API</Name>
                <Host>MYCC-{{ company_code | upper }}-API</Host>
                <Port>8000</Port>
                <Protocol>HTTP</Protocol>
              </WebServer>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: api_webserver_result

    # 4.3 - Clone and Configure WAF Rules
    - name: Get WAF Rule #149 configuration
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Get>
              <WAFPolicy>
                <Filter>
                  <key name="Name" criteria="=">WAF-Rule-149</key>
                </Filter>
              </WAFPolicy>
            </Get>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: waf_149_config
    - name: Clone WAF Rule #149 for APP
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="add">
              <WAFPolicy>
                <Name>WAF-{{ company_code | upper }}-APP</Name>
                <Domain>{{ company_code }}.mycarecrm.com.au</Domain>
                <WebServer>MYCC-{{ company_code | upper }}-APP</WebServer>
                <Status>Enable</Status>
                <!-- Copy other settings from Rule #149 -->
              </WAFPolicy>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: waf_app_result

    - name: Clone WAF Rule #150 for API
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="add">
              <WAFPolicy>
                <Name>WAF-{{ company_code | upper }}-API</Name>
                <Domain>api-{{ company_code }}.mycarecrm.com.au</Domain>
                <WebServer>MYCC-{{ company_code | upper }}-API</WebServer>
                <Status>Enable</Status>
                <!-- Copy other settings from Rule #150 -->
              </WAFPolicy>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: waf_api_result
```

Step 5: Sophos VLAN Interface Configuration

Current Manual Process

1. Go to Network → Select VLAN
2. Click "Add Interface"
3. Select VLAN
4. Configure:
    - **Name:** MYCC-Shortname-vlan
    - **Interface:** Production - Port6
    - **VLAN ID:** [vlan]
    - **IPv4/netmask:** 10.12.[vlan].1/24

Automation Requirements

- ☐ Integrate VLAN creation into Sophos playbook
- ☐ Ensure VLAN is created before VM deployment
- ☐ Configure routing between VLANs

Sophos VLAN Creation

Ansible Playbook (Step 5 - Sophos VLAN):

YAML

```yaml
---
- name: Step 5 - Create Sophos VLAN Interface
  hosts: localhost
  gather_facts: no

  tasks:
    - name: Create VLAN interface on Sophos
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
```

```
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="add">
              <Interface>
                <Name>MYCC-{{ company_code | upper }}-vlan</Name>
                <InterfaceType>VLAN</InterfaceType>
                <ParentInterface>Production - Port6</ParentInterface>
                <VLANID>{{ vlan_id }}</VLANID>
                <IPv4Configuration>Static</IPv4Configuration>
                <IPv4Address>10.12.{{ vlan_id }}.1</IPv4Address>
                <Netmask>255.255.255.0</Netmask>
                <Zone>LAN</Zone>
              </Interface>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
      register: vlan_interface_result

    - name: Enable VLAN interface
      uri:
        url: "{{ sophos_api_url }}/webconsole/APIController"
        method: POST
        body: |
          <Request>
            <Login>
              <Username>{{ sophos_username }}</Username>
              <Password>{{ sophos_password }}</Password>
            </Login>
            <Set operation="update">
              <Interface>
                <Name>MYCC-{{ company_code | upper }}-vlan</Name>
                <Status>Enable</Status>
              </Interface>
            </Set>
          </Request>
        headers:
          Content-Type: "application/xml"
        validate_certs: no
```

Complete Automated Workflow Integration

Updated Task 15 Master Orchestrator

File: `playbooks/task15-master-orchestrator.yml`

New Step Order:

YAML

```yaml
---
- name: Task 15 - Full Stack Deployment with Network Automation
  hosts: localhost
  gather_facts: no

  tasks:
    # Step 0: Credential Management (Azure Key Vault)
    - name: Step 0 - Generate and Store Credentials
      include_tasks: includes/azure-keyvault-integration.yml

    # Step 1: Portal Integration (Manual for now)
    # Payload received from portal webhook

    # Step 2: DNS Configuration (Synergy Wholesale)
    - name: Step 2 - Create DNS Records
      include_tasks: includes/synergy-dns-creation.yml

    # Step 3: UniFi VLAN Assignment
    - name: Step 3 - Configure UniFi VLAN
      include_tasks: includes/unifi-vlan-setup.yml

    # Step 4 & 5: Sophos Firewall Configuration
    - name: Step 4 - Configure Sophos Hosts and WAF
      include_tasks: includes/sophos-firewall-config.yml

    - name: Step 5 - Create Sophos VLAN Interface
      include_tasks: includes/sophos-vlan-creation.yml

    # Step 6: Create VMs on Hyper-V
    - name: Step 6 - Deploy VMs on Hyper-V
      include_tasks: includes/hyperv-vm-creation.yml

    # Step 7-10: Application Deployment
    - name: Step 7 - Deploy Database
      include_tasks: includes/database-deployment.yml

    - name: Step 8 - Deploy API
      include_tasks: includes/api-deployment.yml

    - name: Step 9 - Deploy Application
      include_tasks: includes/app-deployment.yml
    - name: Step 10 - Post-Deployment Validation
      include_tasks: includes/deployment-validation.yml
```

Required AWX Credentials

1. Synergy Wholesale API Credential

```yaml
Credential Type: Custom (Synergy Wholesale API)
Fields:
  - synergy_api_key (secret)
  - sophos_public_ip
```

2. UniFi Controller Credential

```yaml
Credential Type: Custom (UniFi Controller)
Fields:
  - unifi_controller_url
  - unifi_username
  - unifi_password (secret)
```

3. Sophos Firewall Credential

```yaml
Credential Type: Custom (Sophos XG Firewall)
Fields:
  - sophos_api_url
  - sophos_username
  - sophos_password (secret)
```

Information Required to Proceed

Synergy Wholesale

- ☐ API key/credentials
- ☐ API documentation access
- ☐ Sophos public IP address
- ☐ Domain: mycarecrm.com.au

UniFi

- ☐ Controller URL (https://unifi.ui.com or local)
- ☐ Admin username
- ☐ Admin password
- ☐ Site name (default or custom)

Sophos XG Firewall

- ☐ Firewall management IP/URL
- ☐ API username
- ☐ API password
- ☐ Template WAF Rule #149 configuration
- ☐ Template WAF Rule #150 configuration
- ☐ Production Port6 interface details

Deployment Sequence Diagram

Plain Text

```
Portal → AWX Job Launch
  ↓
Step 0: Azure Key Vault (Password Generation)
  ↓
Step 2: Synergy Wholesale DNS (A Records)
  ↓
Step 3: UniFi VLAN Assignment
  ↓
Step 4: Sophos Hosts & WAF Configuration
  ↓
Step 5: Sophos VLAN Interface Creation
  ↓
Step 6: Hyper-V VM Creation
```

```
    ↓
Step 7-9: Application Deployment
    ↓
Step 10: Validation & Testing
    ↓
Deployment Complete → Notify Portal
```

Next Steps for Automation

Immediate (High Priority)

1. **Get API credentials:**
   - Synergy Wholesale API key
   - UniFi Controller credentials
   - Sophos Firewall API credentials
2. **Test API access:**
   - Test Synergy DNS API
   - Test UniFi Controller API
   - Test Sophos Firewall API
3. **Create playbooks:**
   - `includes/synergy-dns-creation.yml`
   - `includes/unifi-vlan-setup.yml`
   - `includes/sophos-firewall-config.yml`
   - `includes/sophos-vlan-creation.yml`

Medium Priority

1. **Create AWX custom credential types**
2. **Add credentials to AWX**
3. **Update Task 15 master orchestrator**
4. **Test each step individually**
5. **Test complete workflow**

Low Priority

1. **Portal webhook integration**

2. **Automated rollback procedures**
3. **Monitoring and alerting**
4. **Documentation and training**

Testing Checklist

Step 2 - DNS

- □ DNS A record created for APP domain
- □ DNS A record created for API domain
- □ DNS resolves correctly from external network
- □ DNS propagation complete

Step 3 - UniFi

- □ VLAN created with correct ID
- □ VLAN subnet configured correctly
- □ VLAN visible in UniFi controller

Step 4 - Sophos Hosts & WAF

- □ APP host object created
- □ DB host object created
- □ API host object created
- □ APP web server created
- □ API web server created
- □ WAF rule for APP created and enabled
- □ WAF rule for API created and enabled

Step 5 - Sophos VLAN

- □ VLAN interface created
- □ VLAN interface enabled
- □ VLAN routing configured

- □ VLAN accessible from other networks

Last Updated: Oct 19, 2025

Document Version: 1.0

Status: Ready for API credential collection and playbook development