## **SOP Template: Secure Container Deployment**

| **Procedure Title** | *Secure-Docker Deployment* |
| --- | --- |
| **SOP ID** | *[CCDC-SOP-001]* |
| **Date** | *[6/04/2025]* |
| **Author** | *Sevag Bairamian* |

### **1. Service Identification**

| **Item** | **Details** |
| --- | --- |
| **Service Name** | *Secure-Docker* |
| **Business Purpose** | *Creates a more secure Docker environment that reduces the attack surface, thus reducing chances of the Docker environment being compromised.* |
| **Data Sensitivity** | *Confidential - stores sensitive business data.* |

### **2. Image Selection & Validation**

| **Rule** | **Specification / Command** | **Justification** |
| --- | --- | --- |
| **1. Use Official Image** | ***Specification:*** *When setting up web servers, it is important to use images that have been verified by docker such as Nginx:latest*  ***Command:*** *docker pull [image name]* | *By using a trusted image, it reduces the attack surface and doesn’t allow us to be using corrupt images.* |
| **2. Pin to Specific Tag/Digest** | ***Specification:*** *Pin it to a specific node such as node:18.17.1-alpine* | *Ensures consistency in terms of images and avoids unexpecting breaking changes that might occur from an unstable image.* |
| **3. Verify Image Provenance** | ***Specification:*** *Use cosign to create key pairs and verify signature.*  ***Command:***cosign verify node :[node version] | *Proves that image is legitimate from other vendors and has not been tampered with in any way.* |

### **3. Runtime Configuration**

| **Rule** | **Specification / Command** | **Justification** |
| --- | --- | --- |
| **1. Run as Non-Root User** | ***Specification:*** Create users for images. Assign them to groups if preferred. | *Not giving users admin privileges reduces chances of privilege escalation attacks* |
| **2. Create Dedicated Network** | ***Specification:*** *Put the docker containers on specific networks.*  ***Command:*** *docker network create [name]* | *Isolates the container traffic to reduce access to other unnecessary services.* |
| **3. Expose Specific Ports** | ***Specification:*** *Expose ports that only need to be used.*  ***Command:*** *docker run -p [port numbers]* | *By choosing which ports to expose, it reduces the amount of ports that can be attacked.* |
| **4. Set Read-Only Filesystem** | ***Specification: -*** *Set read only to true and or run only.*  ***Command****: docker run -d --read-only* | *Prevents hackers or malicious attackers from writing to the disk and making changes if filesystem becomes compromised.* |

### **4. Data Persistence & Volumes**

| **Rule** | **Specification / Command** | **Justification** |
| --- | --- | --- |
| **1. Use Named Volumes** | ***Specification:*** Provide specific names for volumes related to their specific role. For example, name volume db-data if it will be used to store a database of data.  ***Command*** *: docker volume create [volume name]* | so that it is easier to organize and document docker infrastructure. |
| **2. Avoid Host Bind Mounts** | ***Specification:*** Using readonly or ro option to prevent the container from writing to the mount.  **Command**: docker run --read-only | *Protects and maintains integrity of the host file system from being overwritten* |
| **3. Document Volume Purpose** | **Specification:** Use this to store data that can be recovered upon.  **Command:**docker run -v [volume-name]:/[file path]  docker cp [Volume name]:/[file-path] ./[file-name] | *Helps with recovery of data and follows compliance.* |

### **5. Final Deployment Command**

*Combine all the rules above into a complete deployment procedure for your chosen service.*

# [Insert your complete deployment commands here]

# Include network creation, volume setup, and container deployment

# Apply all security controls defined in sections 1-4

# 1. Pull required image

docker pull [image name]

# 2. Create dedicated network

docker network create [network-name]

# 3. Create Volume name

docker volume create [volume-name]

# 4. Use Cosign to verify node

cosign verify node :[node version]

# 5. Run container with proper configuration

docker run -d \

--name [docker name] \

--network [network name] \

-p [port numbers] \

--read-only \

-v [volume-name]:/[file path]\

--user [specify-user]

[image-name]

#6. Verify deployment and test connectivity

docker ps

curl <http://localhost:80>

#7. Backup-data to current directory

docker cp [Volume name]:/[file-path] ./[file-name]

—-------------------------------------------------------------------------------------------------------------

Real life example

# 1. Pull required image

docker pull nginx:1.24-alpine

# 2. Create dedicated network

docker network create secure-network

docker network create --subnet=172.20.0.0/16 secure-network

# 3. Create Volume name

docker volume create [app-data]

# 4. Use Cosign to verify node

cosign verify node :1.24-alpine

# 5. Run container with proper configuration

docker run -d \

--name new-container \

--network secure-container \

-p 443:80 \

--read-only \

-v app-data:/data \

--user 1000:1000

nginx:alpine

#6. Verify deployment and test connectivity

docker ps

curl <http://localhost:80>

# 7. Backup data

docker cp app-data:/etc/nginx/ ./data-backup.conf

### **6. Executive Summary**

**Required:** Provide a 2-3 paragraph executive summary for the CTO highlighting:

The “Secure-Docker” secure container deployment SOP defines stricter security controls to prevent a recurrence of prior misconfiguration-related breaches. The key security controls that are implemented are

* Using well-know base image
* Using signed image verification (Cosign)
* Having strict runtime controls (non-root, read-only
* Having isolated networking
* Backing up data to allow recovery of data

These procedures align with SOC 2 Type II by aligning with least privilege, access control, and system operations. The ISO 27001 Annex A compliance rules that will be followed are

* A.11.2.(privilege management)
* A.11.1.1, Access control Policy.
* A. 10.6.1 - Network Controls
* A.10.5.1- Information Back up
* A.12.5.3 - Restricts on changes to software packages.

The full implementation can be completed over a 4-week schedule if all DevOps, Security and IT teams are involved. Data migration will need to be done during outside-work hours to reduce downtime. Preferred times are during weekends where there are more days allotted to transfer data. Resources required include licenses for Cosign, 2 additional vCPUs, and at least 4 GB RAM. These steps significantly reduce operational risk and improve compliance posture.