



SERVERLESS LAPS WITH INTUNE, FUNCTION APP AND KEY VAULT


SERVERLESS LAPS WITH INTUNE, FUNCTION APP AND KEY VAULT


 [Linkedin \(https://www.linkedin.com/shareArticle?](https://www.linkedin.com/shareArticle?trk=Serverless+LAPS+with+Intune%2C+Function+App+and+Key+Vault&url=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)

[trk=Serverless+LAPS+with+Intune%2C+Function+App+and+Key+Vault&url=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F](https://www.linkedin.com/shareArticle?trk=Serverless+LAPS+with+Intune%2C+Function+App+and+Key+Vault&url=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F))

 [Share \(https://www.facebook.com/sharer.php?u=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F\)](https://www.facebook.com/sharer.php?u=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)

 [Tweet \(https://twitter.com/intent/tweet?](https://twitter.com/intent/tweet?text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault&url=http%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F&via=_Cloud_boy)
[text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault&url=http%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F&via=_Cloud_boy](https://twitter.com/intent/tweet?text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault&url=http%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F&via=_Cloud_boy))

 [Whatsapp \(https://api.whatsapp.com/send?](https://api.whatsapp.com/send?text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)
[text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F](https://api.whatsapp.com/send?text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F))

 [Mail \(mailto:?subject=%20&body=%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F\)](mailto:?subject=%20&body=%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)

This article will describe how setup Serverless LAPS with Intune, Function App and Key Vault.

Situation:

- Full cloud device management (Azure AD Joined devices, Intune managed)
- No LAPS solution, because of no on-premise Active Directory

Microsoft Local Administrator Password Solution (LAPS) is a password manager that utilises Active Directory to manage and rotate passwords for local Administrator accounts across all of your Windows endpoints. LAPS is a great mitigation tool against lateral movement and privilege escalation, by forcing all local Administrator

accounts to have unique, complex passwords, so an attacker compromising one local Administrator account can't move laterally to other endpoints and accounts that may share that same password. A benefit, compared to other password managers, is that LAPS does not require additional computers, or application servers, to manage passwords. The management of these passwords is done entirely through Active Directory components.

SPEAKING ([HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/](https://www.cloud-boy.be/speaking/))
Target:

- Deploying LAPS, serverless, without an on-premise Active Directory.

I've stumbled across this blog post: <https://www.srdn.io/2018/09/serverless-laps-powered-by-microsoft-intune-azure-functions-and-azure-key-vault/> (<https://www.srdn.io/2018/09/serverless-laps-powered-by-microsoft-intune-azure-functions-and-azure-key-vault/>) but it's a bit outdated with all the Azure changes in the last year, so I decided to update it. Credits to John Seerden for the PowerShell scripts though.

1. DEPLOY AN AZURE FUNCTION APP & CONFIGURE IT

In the Azure Portal, navigate to Function Apps and click on 'Add' to create a new Function App. Choose a Subscription and a Resource Group (or create a new one), give your Function App a name and as Runtime stack choose 'PowerShell Core (Preview)'. Click on 'Next'.

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

[Basics](#) [Hosting](#) [Monitoring](#) [Tags](#) [Review](#) [Create](#)

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

MCT Azure Subscription



Resource Group * ⓘ

RG_SLAPS

[Create new](#)

Instance Details

Function App name *

serverlesslaps



.azurewebsites.net

Publish *

Code

Docker Container

Runtime stack *

PowerShell Core (Preview)



Region *

West Europe



Choose or create a new storage account, leave the Operating System setting on 'Windows'. And choose a plan type. In my scenario I choose for Consumption. Click on 'Next'.

Storage



(<https://www.cloud-boy.be>)

[ABOUT ME \(HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/\)](https://www.cloud-boy.be/about-me/)

When creating a function app, you must create or link to a general-purpose Azure Storage account that supports Blobs, Queue, and Table storage.

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Storage account *

(New) storageaccountrgsla8f82

[Create new](#)

Operating system

Windows is the only supported Operating System for your selection of runtime stack.

Operating System *

Linux

Windows

Plan

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

Plan type * ⓘ

Consumption

Choose if you want to enable Application Insights (not necessary) and click on 'Review + Create' After the validation, click on 'Create' to deploy the Function App.

Azure Monitor gives you full observability into your applications, infrastructure, and network. [Learn more](#)

Application Insights

Enable Application Insights *

No

Yes

Application Insights *

(New) serverlesslaps (West Europe)

[Create new](#)

Region

West Europe

Navigate to the function App, and click on 'Platform features'.

Home > Function App > STM-SLAPS

STM-SLAPS
Function Apps

Overview Platform features

Status Running
Availability Loading ...
Subscription MCT Azure Subscription
Subscription ID fc8ca69d-8968-4780-b000-37471d33082f
Resource group RG_SLAPS
URL https://stm-slaps.azurewebsites.net
Location West Europe
App Service plan / pricing tier ASP-RGSLAPS-8690 (Consumption)

Configured features

- Function app settings
- Configuration

Loading ...

Click on 'Identity'.

STM-SLAPS
Function Apps

Search

MCT Azure Subscription

Function Apps

STM-SLAPS

Functions

- Set-KeyVaultSecret
- Integrate
- Manage
- Monitor

Proxies

Slots

PowerShell Functions are a preview offer. [Learn more](#)

Overview Platform features

Search features

General Settings

- Function app settings
- Configuration
- Properties
- Backups
- All settings

Code Deployment

- Container settings

Development tools

- Logic Apps
- Console (CMD / PowerShell)
- Advanced tools (Kudu)
- App Service Editor
- Resource Explorer
- Site Extensions

Networking

- Networking
- SSL
- Custom domains
- Authentication / Authorization
- Identity
- Push notifications

Monitoring

- Dagnostic logs
- Log streaming
- Process explorer
- Metrics

API

- API Management
- API definition
- CORS

App Service plan

- App Service plan
- Scale up
- Scale out
- Quotas

Resource management

- Diagnose and solve problems
- Activity log
- Access control (IAM)
- Tags
- Locks
- Export template

In 'System Assigned' switch the status to 'On'. Click on 'Save'.

A system assigned managed identity enables Azure resources to authenticate to cloud services (e.g. Azure Key Vault) without storing credentials in code. We'll grant this managed identity access to our Key Vault later on.

System assigned User assigned

A system assigned managed identity is a cloud resource that can be used to authenticate to cloud services (e.g. Azure Key Vault) without the need for credentials. It is created automatically and has a permission resource. Additionally, each resource (e.g. Virtual Machine) can only have one system assigned managed identity. [Learn more about Managed identities.](#)

Save Discard Refresh Got feedback?

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Status ⓘ

Off On

Object ID ⓘ

42660ffd-b954-4cb4-8cd3-523770cbc70c



This resource is registered with Azure Active Directory. You can control its access to services like Azure Resource Manager, Azure Key Vault, etc. [Learn more](#)

Our Function App also requires a minimum TLS version of 1.2, so go back to the 'Platform Features' and click on 'SSL'.

STM-SLAPS

Function Apps

Search

MCT Azure Subscription

Function Apps

STM-SLAPS

Functions

Set-KeyVaultSecret

Integrate

Manage

Monitor

Proxies

Slots

PowerShell Functions are a preview offer. [Learn more](#)

Overview Platform features

Search features

General Settings

Function app settings

Configuration

Properties

Backups

All settings

Code Deployment

Container settings

Development tools

Logic Apps

Console (CMD / PowerShell)

Advanced tools (Kudu)

App Service Editor

Resource Explorer

Site Extensions

Networking

Networking

SSL

Custom domains

Authentication / Authorization

Identity

Push notifications

Monitoring

D diagnostic logs

Log streaming

Process explorer

Metrics

API

API Management

API definition

CORS

App Service plan

App Service plan

Scale up

Scale out

Quotas

Resource management

Diagnose and solve problems

Activity log

Access control (IAM)

Tags

Locks

Export template

Set HTTPS Only to 'On' and select Minimum TLS Version of '1.2'. Click on 'Refresh'.



Protocol Settings

Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only: ⓘ

Off

On

Minimum TLS Version ⓘ

1.0

1.1

1.2

Incoming client certificates ⓘ

Off

On

2. DEPLOY AN AZURE KEY VAULT AND GRANT OUR MANAGED SERVICE IDENTITY ACCESS

In the Azure Portal, navigate to Key Vaults and click on 'Add' to create a new Key Vault. Choose a Subscription and a Resource Group (or create a new one), give your Key Vault a name and leave the pricing tier on 'Standard'. Click on 'Next'.



(https://www.cloud-boy.be)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules (HSMs) with FIPS 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance. [Learn more](#)

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

MCT Azure Subscription



Resource group *

RG_SLAPS

[Create new](#)

Instance details

Key vault name * ⓘ

serverlesslaps



Region *

West Europe



Pricing tier * ⓘ

Standard



Click on 'Add Access Policy'.

Enable Access to:

- ☐ Azure Virtual Machines for deployment ⓘ
- ☐ Azure Resource Manager for template deployment ⓘ
- ☐ Azure Disk Encryption for volume encryption ⓘ

+ Add Access Policy

Current Access Policies

Name	Category	Email	Key Permissions
USER			
 Tim Hermie	USER	tim.hermie@switchtomodern.be	9 selected

Select 'Set' in secret permissions. Afterwards click on 'Select principal'.

[Home](#)

[Key vaults](#)


[Create key vault](#)

[Add access policy](#)

BLOG (HTTPS://WWW.CLOUD-BOY.BE/)

SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)



(https://www.cloud-boy.be)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

Add access policy

Configure from template (optional)

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Key permissions

Secret permissions

Certificate permissions

Select principal

Authorized application ⓘ

0 selected

Set

0 selected

*

None selected

>

None selected


🔒

Add

Here we will choose our newly made Funtion App (which we gave a system assigned managed identity). Select the principal & click on 'Add'.

Principal

Select a principal



(https://www.cloud-boy.be)


SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

Select

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

STM-SLAPS



STM-SLAPS

You'll see this screen next when it's done right:

Basics

Access policy

Virtual network

Tags

Review + create

Enable Access to:

☐

Azure Virtual Machines for deployment

☐



Azure Resource Manager for template deployment

☐

Azure Disk Encryption for volume encryption

+ Add Access Policy

Current Access Policies

Name	Category	Email	Key Permissions	Secret Permissions	Cer
APPLICATION					
<div> STM-SLAPS</div>	APPLICATION		0 selected	Set	0
USER					
<div> Tim Hermie</div>	USER	tim.hermie@switchtomodern.be	9 selected	7 selected	15

Click on 'Review + Create' and after validation click on 'Create'.

[BLOG \(HTTPS://WWW.CLOUD-BOY.BE/\)](https://www.cloud-boy.be/)

[SHORTS \(HTTPS://WWW.CLOUD-BOY.BE/SHORTS/\)](https://www.cloud-boy.be/shorts/)



[\(https://www.cloud-boy.be\)](https://www.cloud-boy.be/)

[ABOUT ME \(HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/\)](https://www.cloud-boy.be/about-me/)

3. CREATE AND TEST THE AZURE FUNCTION

[SPEAKING \(HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/\)](https://www.cloud-boy.be/speaking/)

Go back to your Function App. Select the Function and click on 'New Function'.

[Home](#) > [Function App](#) > STM-SLAPS

STM-SLAPS

Function Apps

🔍 "STM-SLAPS" ✖

MCT Azure Subscription ▼

☰ Function Apps

▼ ⚡ STM-SLAPS

▼ ☰ Functions +

+ New function

f

Functions

🔍 Search functions

NAME ▼

STATUS ▼

Choose 'HTTP Trigger'.

Choose a template below or [go to the quickstart](#)

🔍 Search by trigger, language, or description

Scenario: All ▼

HTTP

HTTP trigger

A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string

🕒

Timer trigger

A function that will be run on a specified schedule

✉

Azure

A function that will be run on a specified schedule

📧

Azure Service Bus Queue trigger

A function that will be run whenever a message is added to a specified Service Bus queue

📧

Azure Service Bus Topic trigger

A function that will be run whenever a message is added to the specified Service Bus Topic

📧

Azure

A function that will be run whenever a message is added to the specified cont

📡

Azure Event Hub trigger

🌌

Azure Cosmos DB trigger

📡

IoT H

Give it the name 'Set-KeyVaultSecret'. Authorisation level is 'Function'. Click on 'Create'.

[BLOG \(HTTPS://WWW.CLOUD-BOY.BE/\)](https://www.cloud-boy.be/)

[SHORTS \(HTTPS://WWW.CLOUD-BOY.BE/SHORTS/\)](https://www.cloud-boy.be/shorts/)

 HTTP trigger



[\(https://www.cloud-boy.be\)](https://www.cloud-boy.be/)

[ABOUT ME \(HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/\)](https://www.cloud-boy.be/about-me/)

New Function
[SPEAKING \(HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/\)](https://www.cloud-boy.be/speaking/)

Name:

Set-KeyVaultSecret

HTTP trigger

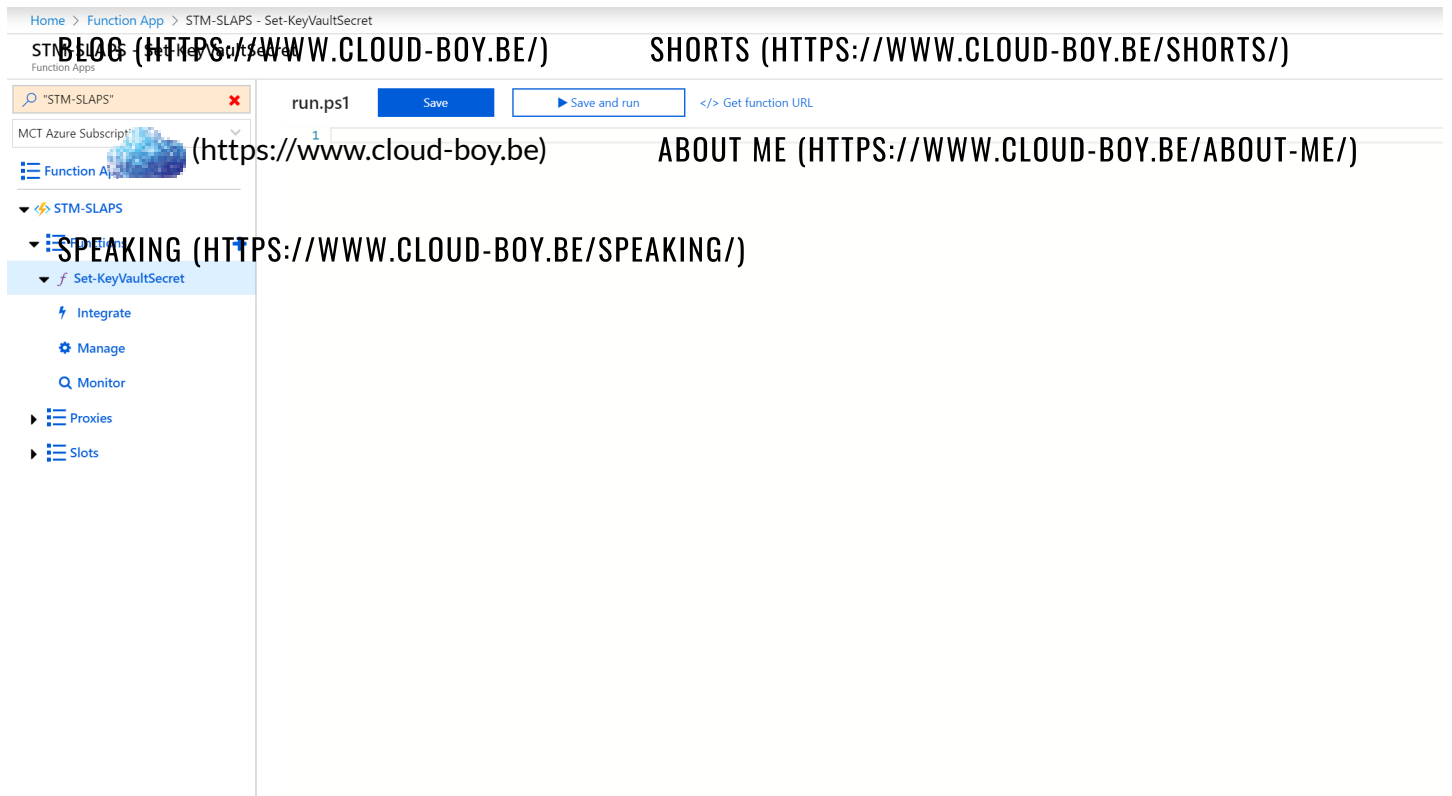
Authorization level ⓘ

Function

Create

Cancel

Once deployed, click on your 'Set-KeyVaultSecret' function and delete all the code. You'll function will be blank now:



Download Set-KeyVaultSecret.ps1 from <https://github.com/jseerden/SLAPS> (https://github.com/jseerden/SLAPS) and insert the code in your Function App. Edit the \$keyVaultName variable with the name of your Key Vault. Click on 'Save'.

STM-SLAPS - Set-KeyVaultSecret

Function

run.ps1

Save

Run

</> Get function URL

MCT Azure Subscription

Function Apps

STM-SLAPS

Functions

Set-KeyVaultSecret

Manage

Monitor

Proxies

Slots

```

1 using namespace System.Net
2
3 param(
4     [Parameter(Mandatory = $true)]
5     string $keyVaultName
6 )
7
8 $keyVaultName = "STM-SLAPS"
9
10 # Azure Key Vault resource to obtain access token
11 $vaultSecretUri = "https://$keyVaultName.vault.azure.net/secrets/$($request.body.keyName)/?api-version=2016-10-01"
12
13 # Get Azure Key Vault Access Token using the Function's Managed Service Identity
14 $authToken = Invoke-RestMethod -Method Get -Headers @{ 'Secret' = $env:MSI_SECRET } -Uri "$($env:MSI_ENDPOINT)?resource=$vaultSecretUri&api-version=$apiVersion"
15
16 # Use Azure Key Vault Access Token to create Authentication Header
17 $authHeader = @{ Authorization = "Bearer $($authToken.access_token)" }
18
19 # Generate a random password
20 function New-Password {
21     $alphabets = 'a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z'
22     $numbers = 2..9
23     $specialCharacters = '!,@,#,$,%,&,*?,+&'
24     $sarray = @()
25     $sarray += $alphabets.Split(',') | Get-Random -Count 10
26     $sarray[0] = $sarray[0].ToUpper()
27     $sarray[-1] = $sarray[-1].ToUpper()
28     $sarray += $numbers | Get-Random -Count 3
29     $sarray += $specialCharacters.Split(',') | Get-Random -Count 3
30     ($sarray | Get-Random -Count $sarray.Count) -join ""
31 }
32
33 $password = New-Password
34
35 # Generate a new body to set a secret in the Azure Key Vault
36 $body = $request.body | Select-Object -Property * -ExcludeProperty keyName
37
38 # Append the random password to the new body
39 $body | Add-Member -NotePropertyName value -NotePropertyValue "$password"
40
41 # Convert the body to JSON
42 $body = $body | ConvertTo-Json
43
44 # Azure Key Vault Uri to set a secret
45 $vaultSecretUri = "https://$keyVaultName.vault.azure.net/secrets/$($request.body.keyName)/?api-version=2016-10-01"
46
47 # Set the secret in Azure Key Vault
48 $null = Invoke-RestMethod -Method PUT -Body $body -Uri $vaultSecretUri -ContentType 'application/json' -Headers $authHeader -ErrorAction Stop
49
50 # Return the password in the response
51 Push-OutputBinding -Name Response -Value ([HttpResponseContext]@{
52     Body = $password
53 })
54
55 }
```

Now you can test the function by clicking on ‘Test’. Add the following code in the ‘Request body’ field. Click on ‘Run’.

```
{
  "keyName": "TEST-PC01",
  "contentType": "Local Administrator Credentials",
  "tags": {
    "Username": "localadmin"
  }
}
```

Be aware, if you copy paste from my site you have to replace the ” with new ones inside the Function App, otherwise you’ll get errors.

BLOG (HTTPS://WWW.CLOUD-BOY.BE/)

SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

run.ps1 Save Run </> Get function URL

```
1 using namespace System.Net
2
3 param (
4     [string] $url = "https://www.cloud-boy.be"
5 )
6
7
8 $keyVaultName = "STM-SLAPS"
9
10 # Azure Key Vault resource to obtain access token
11 $vaultUri = "https://$keyVaultName.vault.azure.net/"
12
13
14 # Get Azure Key Vault Access Token using the Function's Managed Service Identity
15 $authToken = Invoke-RestMethod -Method Get -Headers @{ 'Secret' = $env:MSI_SECRET } -Uri "$($env:MSI_ENDPOINT)?resource=$vaultUri&api-version=2016-10-01"
16
17 # Use Azure Key Vault Access Token to create Authentication Header
18 $authHeader = @{ Authorization = "Bearer $($authToken.access_token)" }
19
20 # Generate a random password
21 function New-Password {
22     $alphabets = 'a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z'
23     $numbers = 2..9
24     $specialCharacters = '!,@,#,$,%,&,*?,'
25     $array = @()
26     $array += $alphabets.Split(',') | Get-Random -Count 10
27     $array[0] = $array[0].ToUpper()
28     $array[-1] = $array[-1].ToUpper()
29     $array += $numbers | Get-Random -Count 3
30     $array += $specialCharacters.Split(',') | Get-Random -Count 3
31     ($array | Get-Random -Count $array.Count) -join ""
32 }
33
34 $password = New-Password
35
36 # Generate a new body to set a secret in the Azure Key Vault
37 $body = $request.body | Select-Object -Property * -ExcludeProperty keyName
38
39 # Append the random password to the new body
40 $body | Add-Member -NotePropertyName value -NotePropertyValue "$password"
41
42 # Convert the body to JSON
43 $body = $body | ConvertTo-Json
44
45 # Azure Key Vault Uri to set a secret
46 $vaultSecretUri = "https://$keyVaultName.vault.azure.net/secrets/$($request.Body.keyName)?api-version=2016-10-01"
47
48 # Set the secret in Azure Key Vault
49 $null = Invoke-RestMethod -Method PUT -Body $body -Uri $vaultSecretUri -ContentType 'application/json' -Headers $authHeader -ErrorAction Stop
50
51 # Return the password in the response
52 Push-OutputBinding -Name Response -Value ([HttpResponseContext]@{
53     Body = $password
54 })
55 }
```

(https://www.cloud-boy.be)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

View files Test

HTTP method

POST

There are no query parameters

+ Add parameter

Headers

There are no headers

+ Add header

Request body

```
1 {
2     "keyName": "TEST-PC01",
3     "contentType": "Local Administrator Credentials"
4     "tags": {
5         "Username": "localadmin"
6     }
7 }
```

Output

Navigate to your Key vault and check if the local admin credentials are stored there. Click on TEST-PC01.

Home > Key vaults > STM-SLAPS - Secrets

Key vaults STM-SLAPS - Secrets

Switch To Modern

+ Add Edit columns More

Filter by name...

☐ Name ↑↓

☐ STM-SLAPS

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Keys Secrets Certificates Access policies Firewalls and virtual networks Properties Locks Export template Monitoring Alerts Metrics Diagnostic settings Logs Support + troubleshooting Resource health New support request

Search (Ctrl+/)

+ Generate/Import Refresh Restore Backup

Name	Type	Status	Expiration Date
SWITCH-7605	Local Administrator Credentials	✓ Enabled	
TEST-PC01	Local Administrator Credentials	✓ Enabled	

Click on the current version of TEST-PC01

Home > Key vaults > STM SLAPS - Secrets > TEST-PC01

BLOG (HTTPS://WWW.CLOUD-BOY.BE/)

SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

TEST-PC01

Versions

(https://www.cloud-boy.be)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

+ New Version

Refresh

Delete

Download Backup

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Status

CURRENT VERSION


d676fe674ed24233922765e254cdfa40



✓ Enabled

Click on 'Show secret value', your local admin password is stored there:

Home > Key vaults > STM-SLAPS > Secrets > TEST-PC01 > d676fe674ed24233922765e254cdfa40

[BLOG \(HTTPS://WWW.CLOUD-BOY.BE/\)](https://www.cloud-boy.be/) [SHORTS \(HTTPS://WWW.CLOUD-BOY.BE/SHORTS/\)](https://www.cloud-boy.be/shorts/)

 **d676fe674ed24233922765e254cdfa40**
(<https://www.cloud-boy.be/>) [ABOUT ME \(HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/\)](https://www.cloud-boy.be/about-me/)

 Save  Discard


SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Properties

Created 10/28/2019, 7:39:09 PM

Updated 10/28/2019, 7:39:09 PM

Secret Identifier

`https://stm-slaps.vault.azure.net/secrets/TEST-PC01/d676fe674ed24233922765e254cdf...` 

Settings

Set activation date? ⓘ ☐

Set expiration date? ⓘ ☐

Enabled? Yes No

Tags


1 tag >

Secret

Content type (optional)

`Local Administrator Credentials`

Hide Secret Value

`g2zHCqw&m*jvt67` 

4. DEPLOY THE POWERSHELL SCRIPT WITH INTUNE

First we need the Function App URL. Navigate to your SetKeyVaultSecret Function App. Click on 'Get Function URL'.

Home > Function App > STM-SLAPS - Set-KeyVaultSecret

STM-SLAPS - Set-KeyVaultSecret
Function Apps

BLOG (HTTPS://WWW.CLOUD-BOY.BE/) SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

(HTTPS://WWW.CLOUD-BOY.BE) ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

run.ps1

```

1 using namespace System.Net
2
3 param(
4     [Parameter(Mandatory = $true)]
5     $Request
6 )
7
8 $keyVaultName = "STM-SLAPS"
9
10 # Azure Key Vault resource to obtain access token
11 $vaultTokenUri = 'https://vault.azure.net'
12 $apiVersion = '2017-09-01'
13
14 # Get Azure Key Vault Access Token using the Function's Managed Service Identity
15 $authToken = Invoke-RestMethod -Method Get -Headers @{ 'Secret' = $env:MSI_SECRET } -Uri "($env:MSI_ENDPOINT)?resource=$vaultTokenUri&api-version=$apiVersion"
16
17 # Use Azure Key Vault Access Token to create Authentication Header
18 $authHeader = @{ Authorization = "Bearer $($authToken.access_token)" }
19
20 # Generate a random password
21 function New-Password {
22     $alphabets = 'a,b,c,d,e,f,g,h,i,j,k,m,n,p,q,r,t,u,v,w,x,y,z'
23     $numbers = 2..9
24     $specialCharacters = '!,@,#,$,%,&,*?,+'
25     $array = @()
26     $array += $alphabets.Split(',') | Get-Random -Count 10
27     $array[0] = $array[0].ToUpper()
28     $array[-1] = $array[-1].ToUpper()
29     $array += $numbers | Get-Random -Count 3
30     $array += $specialCharacters.Split(',') | Get-Random -Count 3
31 }

```

Click on 'Copy'.

Home > Function App > STM-SLAPS - Set-KeyVaultSecret

STM-SLAPS - Set-KeyVaultSecret
Function Apps

run.ps1

```

1 using namespace System.Net
2
3 param(
4     [Parameter(Mandatory = $true)]
5     $Request
6 )
7
8 $keyVaultName = "STM-SLAPS"
9
10 # Azure Key Vault resource to obtain access token
11 $vaultTokenUri = 'https://vault.azure.net'
12 $apiVersion = '2017-09-01'
13
14 # Get Azure Key Vault Access Token using the Function's Managed Service Identity
15 $authToken = Invoke-RestMethod -Method Get -Headers @{ 'Secret' = $env:MSI_SECRET } -Uri "($env:MSI_ENDPOINT)?resource=$vaultTokenUri&api-version=$apiVersion"
16
17 # Use Azure Key Vault Access Token to create Authentication Header
18 $authHeader = @{ Authorization = "Bearer $($authToken.access_token)" }
19
20 # Generate a random password
21 function New-Password {
22     $alphabets = 'a,b,c,d,e,f,g,h,i,j,k,m,n,p,q,r,t,u,v,w,x,y,z'
23     $numbers = 2..9
24     $specialCharacters = '!,@,#,$,%,&,*?,+'
25     $array = @()

```

Get function URL

Key	URL
default (Function key)	OSV0fm2gaBN9t8udHEwdDhKgpdg/gH39cJYCGPlyrbhIVUgCw==

Copy

Download New-LocalAdmin.ps1 from <https://github.com/jseerden/SLAPS> (https://github.com/jseerden/SLAPS) and edit the following variables:

\$uri = 'PASTE URL HERE'

Save the .ps1 file.

Navigate to the Intune dashboard (<https://devicemanagement.microsoft.com/>) (https://devicemanagement.microsoft.com/). Go to Devices – PowerShell scripts. Click on 'Add' to upload our New-LocalAdmin.ps1 script.

Microsoft 365 Device Management

Home

Dashboards

All services

My data

Devices

Apps

Endpoint security

Users

Groups

Tenant administration

Troubleshooting + support

Dashboard

Devices - PowerShell scripts

Devices - PowerShell scripts

Search (Ctrl+/)

PowerShell scripts

All devices

Monitor

By platform

Windows

iOS

macOS

Android

Device enrollment

Enroll devices

Policy

Compliance policies

Conditional access

Configuration profiles

PowerShell scripts

Device security

Windows 10 update rings

Update policies for iOS

Enrollment restrictions

eSIM cellular profiles (preview)

Policy sets

Other

Device clean-up rules

Device categories

Help and support

Help and support

Script Name

7Zip Install

Adobe Reader DC Install

Belgium eID Middleware

Belgium eID Viewer

Bluebeam Install

CCCleaner Install

Chocolatey Agent Install

Chocolatey Auto Upgrade All

Chrome Install

Citrix Workspace Install

Crystal Reports Runtime Install

CutePDF Install

Discord Install

Edge Dev Insider Install

Firefox Install

Greenshot Install

IrfanView Install

Java Install

Keepass Install

mIRC Install

Notepad++ Install

PDFSam Install

SketchupMake Install

SketchupViewer Install

Spotify Install

uTorrent Install

Visual Studio Code Install

VLC Install

W10 - Clear WSUS Settings

W10 - Enable Sandbox

W10 - SLAPS

W10 - Upload Windows Autopilot Device Information

WhatsApp Install

[BLOG \(HTTPS://WWW.CLOUD-BOY.BE/\)](https://www.cloud-boy.be/)

[SHORTS \(HTTPS://WWW.CLOUD-BOY.BE/SHORTS/\)](https://www.cloud-boy.be/shorts/)

[ABOUT ME \(HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/\)](https://www.cloud-boy.be/about-me/)

[SPEAKING \(HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/\)](https://www.cloud-boy.be/speaking/)

Name your script and click on ‘Next’.



SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

Name ✓

Description

Choose your modified .ps1 script and leave the 3 settings on 'No'.

Script location * ⓘ

Run this script using the logged on credentials ⓘ Yes No

Enforce script signature check ⓘ Yes No

Run script in 64 bit PowerShell Host ⓘ Yes No

Deploy it to your testgroup. And follow up. You should see that the script got deployed successfully to your target device.

Dashboard > Devices - PowerShell scripts > W10 - SLAPS

BLOG (HTTPS://WWW.CLOUD-BOY.BE/)
SHORTS (HTTPS://WWW.CLOUD-BOY.BE/SHORTS/)

W10 - SLAPS
Windows 10 and later

Search
(HTTPS://WWW.CLOUD-BOY.BE/)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

Overview
Manage
Properties
Monitor
Device status
User status

Created: : 10/28/19, 07:42:36 PM

Status for checked-in devices

1
DEVICES

Succeeded
1
Error
0

And if you check again in your Azure Key Vault, the local admin password of your device should be there too:

Home > Key vaults > STM-SLAPS - Secrets

Key vaults
Switch To Modern

STM-SLAPS - Secrets
Key vault

+ Add
Edit columns
More

Filter by name...

Name
STM-SLAPS

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings
Keys
Secrets
Certificates
Access policies
Firewalls and virtual networks
Properties
Locks
Export template

+ Generate/Import
Refresh
Restore Backup

Name	Type	Status
SWITCH-7605	Local Administrator Credentials	✓ Enabled
TEST-PC01	Local Administrator Credentials	✓ Enabled

Happy testing!



(https://www.cloud-boy.be)

ABOUT ME (HTTPS://WWW.CLOUD-BOY.BE/ABOUT-ME/)

device-hashes-from-hp-for-autopilot-pre-production-testing (https://www.cloud-boy.be/blog/get-device-hashes-from-hp-for-autopilot-pre-production-testing/)

SPEAKING (HTTPS://WWW.CLOUD-BOY.BE/SPEAKING/)

- Run as admin gives black screen in Quick Assist/TeamViewer – Intune fix (https://www.cloud-boy.be/blog/run-as-admin-gives-black-screen-in-quick-assist-teamviewer-intune-fix/)
- Intune – change Primary User of a device (https://www.cloud-boy.be/blog/intune-change-primary-user-of-a-device/)
- Ransomware protection (Controlled Folder Access) setup with Intune (https://www.cloud-boy.be/blog/ransomware-protection-controlled-folder-access-setup-with-microsoft-endpoint-manager/)
- Windows Hello for Business multi-factor unlock with Intune (https://www.cloud-boy.be/blog/windows-hello-for-business-multi-factor-unlock-with-mem-intune/)



Linkedin (https://www.linkedin.com/shareArticle?

trk=Serverless+LAPS+with+Intune%2C+Function+App+and+Key+Vault&url=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)



Share (https://www.facebook.com/sharer.php?u=https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)



Tweet (https://twitter.com/intent/tweet?

text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault&url=http:boy.be/blog/serverless-laps-with-intune-function-app-and-key-vault/&via=_Cloud_boy)



Whatsapp (https://api.whatsapp.com/send?

text=Serverless%20LAPS%20with%20Intune%2C%20Function%20App%20and%20Key%20Vault%20https%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)



Mail (mailto:?subject=%20&body=%20https%3A%2F%2Fwww.cloud-boy.be%2Fblog%2Fserverless-laps-with-intune-function-app-and-key-vault%2F)



PREVIOUS POST

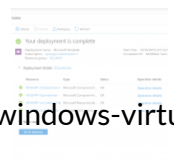
Silently remove SCCM client and enroll device in Intune
(https://www.cloud-boy.be/blog/silently-remove-sccm-client-and-enroll-device-in-intune/)



(https://www.cloud-boy.be/blog/windows-virtual-deployment-template-deployment-is-not-valid/)

NEXT POST

WVD – The template deployment not valid
(https://www.cloud-boy.be/blog/windows-virtual-deployment-template-deployment-is-not-valid/)



sccm-client-and-enroll-device-in-intune/)

(https://www.cloud-boy.be/)



©2020 - Cloud-Boy - Adventures in EMS world.