Acronis

Acronis Cyber Cloud

Integration with Ninja RMM

INTEGRATION GUIDE

Revision: 9/16/2021

Table of contents

1 Introduction

This document describes how to enable and configure the integration of Acronis Cyber Cloud with Ninja RMM.

Once setup, the integration enables you to:

- Deploy Acronis on Windows, Linux and Mac devices
- Monitor protected devices
- Apply and revoke protection plans
- Run the following types of automated tasks - backup, antivirus scan, malware scan, vulnerability assessment and patch management
- Uninstall Acronis agents from Windows, Linux and Mac devices

All this functionality is available from within Ninja RMM, without having to go to the Acronis Cyber Protect web interface.

2 Glossary

- **MSP** - A Managed Service Provider, who uses both Ninja RMM and Acronis Cyber Protect
- **Customer** - A client of the MSP
- **Partner tenant** - the MSP account on Acronis Cyber Cloud
- **Customer tenant** - the customer account on Acronis Cyber Cloud

3 Prerequisites

To use this integration, you should have:

- At least a single, fully configured Ninja RMM account
- An Acronis Cyber Cloud account with:
    - at least one customer tenant and one user, setup with Acronis administration permissions
    - at least one protection plan, configured to be used as the default one

4 How the integration works

The solution consists of a set of scripts run on workloads.

Those scripts are downloaded from Acronis, uploaded to and scheduled from Ninja RMM.

If anywhere throughout this document, you have to provide a registration token, here are the steps to obtain it:

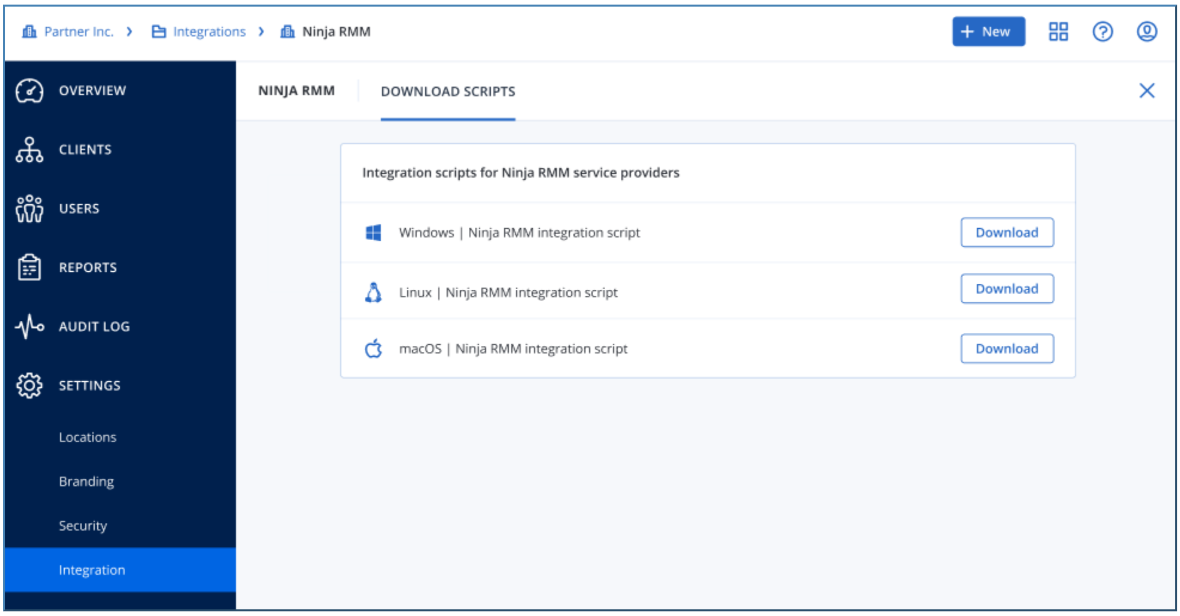
1. Log in to the Acronis Cyber Protection console.
2. Click **Add Device** and scroll down to **Registration token**. Then click **GENERATE**.
3. Select a token with a maximum lifetime value and click **GENERATE TOKEN**.
4. Copy the token you just generated.

5 Setup

In order to set up an Acronis integration for Ninja RMM, do the following:

1. Go to **Acronis Cyber Protect Cloud Management Console** > **Settings** > **Integrations**.
2. Click on the **Ninja RMM** tile.

3. Download the **Ninja RMM Scripts** archive file from the **Download Scripts** tab.

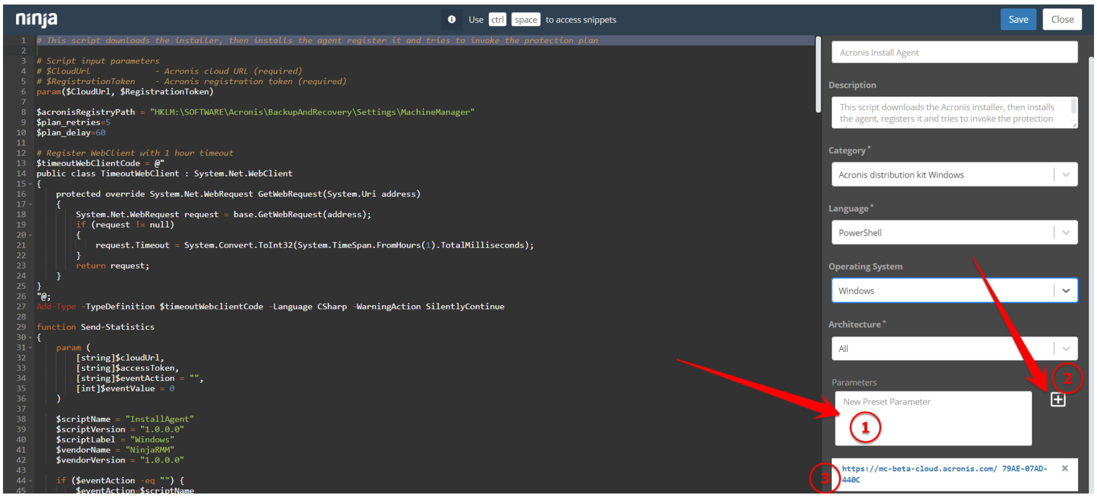


Integration scripts for Ninja RMM service providers

Windows | Ninja RMM integration script	Download
Linux | Ninja RMM integration script	Download
macOS | Ninja RMM integration script	Download

6 Upload scripts to Ninja RMM

1. Go to **Ninja RMM** > **Configuration** > **Scripting** > **Import New Script**.
2. Locate the scripts downloaded from Acronis and upload individually the ones of your choice.
3. Next:
   a. Give a proper name (similar to the script's file name)
   b. Enter a description
   c. Assign to a category
   d. Select the language of the imported script (PowerShell for Windows and ShellScript for Linux)
   e. Select an operating system
   f. Select architecture (**All** is recommended)
   g. For each of the following scripts:
      i. **Acronis_install_agent**: in the **New Preset Parameter** field, add a parameter by entering the currently used Acronis Datacenter URL, followed by a blank space and Acronis registration token (see "How the integration works" (p. 6)). Click on the **+** button to create a parameter. The newly created parameter should resemble the following example: "https://mc-beta-cloud.acronis.com 79AE-07AD-440C"



```
# This script downloads the installer, then installs the agent register it and tries to invoke the protection plan

# Script input parameters
# $CloudUrl             - Acronis cloud URL (required)
# $RegistrationToken    - Acronis registration token (required)
param($CloudUrl, $RegistrationToken)

$acronisRegistryPath = "HKLM:\SOFTWARE\Acronis\BackupAndRecovery\Settings\MachineManager"
$plan_retries=5
$plan_delay=60

# Register WebClient with 1 hour timeout
$timeoutWebClientCode = @"
public class TimeoutWebClient : System.Net.WebClient
{
    protected override System.Net.WebRequest GetWebRequest(System.Uri address)
    {
        System.Net.WebRequest request = base.GetWebRequest(address);
        if (request != null)
        {
            request.Timeout = System.Convert.ToInt32(System.TimeSpan.FromHours(1).TotalMilliseconds);
        }
        return request;
    }
}
"@;
Add-Type -TypeDefinition $timeoutWebclientCode -Language CSharp -WarningAction SilentlyContinue

function Send-Statistics
{
    param (
        [string]$cloudUrl,
        [string]$accessToken,
        [string]$eventAction = "",
        [int]$eventValue = 0
    )

    $scriptName = "InstallAgent"
    $scriptVersion = "1.0.0.0"
    $scriptLabel = "Windows"
    $vendorName = "NinjaRMM"
    $vendorVersion = "1.0.0.0"

    if ($eventAction -eq "") {
        $eventAction=$scriptName
```

Acronis Install Agent

Description: This script downloads the Acronis installer, then installs the agent, registers it and tries to invoke the protection

Category *: Acronis distribution kit Windows

Language *: PowerShell

Operating System: Windows

Architecture *: All

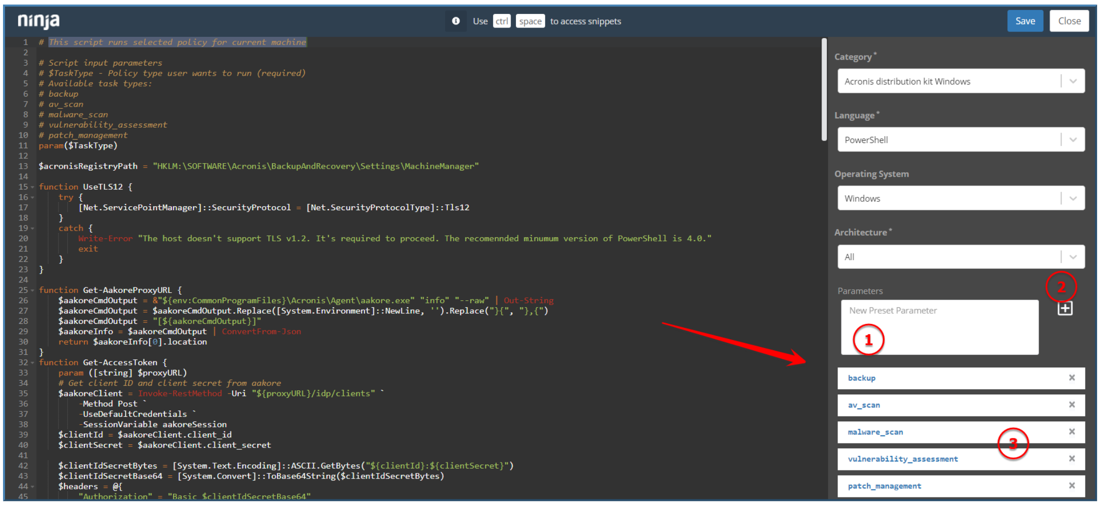
Parameters: New Preset Parameter

https://mc-beta-cloud.acronis.com/ 79AE-07AD-440C

Click **Save**, then **Close**.

      ii. **Acronis_scans**: add the following 5 individual parameters:
         - backup
         - av_scan
         - malware_scan
         - vulnerability_assessment

- patch_management



```
# This script runs selected policy for current machine

# Script input parameters
# $TaskType - Policy type user wants to run (required)
# Available task types:
# backup
# av_scan
# malware_scan
# vulnerability_assessment
# patch_management
param($TaskType)

$acronisRegistryPath = "HKLM:\SOFTWARE\Acronis\BackupAndRecovery\Settings\MachineManager"

function UseTLS12 {
    try {
        [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    }
    catch {
        Write-Error "The host doesn't support TLS v1.2. It's required to proceed. The recomennded minumum version of PowerShell is 4.0."
        exit
    }
}

function Get-AakoreProxyURL {
    $aakoreCmdOutput = &"${env:CommonProgramFiles}\Acronis\Agent\aakore.exe" "info" "--raw" | Out-String
    $aakoreCmdOutput = $aakoreCmdOutput.Replace([System.Environment]::NewLine, '').Replace("}{", "},{")
    $aakoreCmdOutput = "[${aakoreCmdOutput}]"
    $aakoreInfo = $aakoreCmdOutput | ConvertFrom-Json
    return $aakoreInfo[0].location
}
function Get-AccessToken {
    param ([string] $proxyURL)
    # Get client ID and client secret from aakore
    $aakoreClient = Invoke-RestMethod -Uri "${proxyURL}/idp/clients" `
        -Method Post `
        -UseDefaultCredentials `
        -SessionVariable aakoreSession
    $clientId = $aakoreClient.client_id
    $clientSecret = $aakoreClient.client_secret

    $clientIdSecretBytes = [System.Text.Encoding]::ASCII.GetBytes("${clientId}:${clientSecret}")
    $clientIdSecretBase64 = [System.Convert]::ToBase64String($clientIdSecretBytes)
    $headers = @{
        "Authorization" = "Basic $clientIdSecretBase64"
```

Category *: Acronis distribution kit Windows

Language *: PowerShell

Operating System: Windows

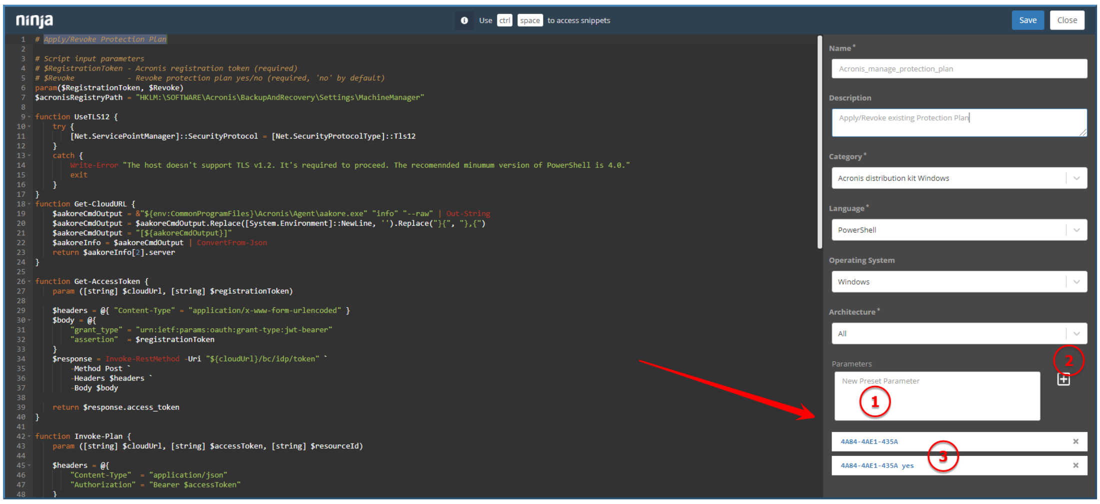
Architecture *: All

Parameters: New Preset Parameter

- backup ×
- av_scan ×
- malware_scan ×
- vulnerability_assessment ×
- patch_management ×

iii. **Acronis_manage_protection_plan**: add the following 2 separate parameters:

- Acronis registration token (see "How the integration works" (p. 6))
- Acronis registration token, followed by a blank space and "**yes**"



```
# Apply/Revoke Protection Plan

# Script input parameters
# $RegistrationToken - Acronis registration token (required)
# $Revoke            - Revoke protection plan yes/no (required, 'no' by default)
param($RegistrationToken, $Revoke)
$acronisRegistryPath = "HKLM:\SOFTWARE\Acronis\BackupAndRecovery\Settings\MachineManager"

function UseTLS12 {
    try {
        [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    }
    catch {
        Write-Error "The host doesn't support TLS v1.2. It's required to proceed. The recomennded minumum version of PowerShell is 4.0."
        exit
    }
}
function Get-CloudURL {
    $aakoreCmdOutput = &"${env:CommonProgramFiles}\Acronis\Agent\aakore.exe" "info" "--raw" | Out-String
    $aakoreCmdOutput = $aakoreCmdOutput.Replace([System.Environment]::NewLine, '').Replace("}{", "},{")
    $aakoreCmdOutput = "[${aakoreCmdOutput}]"
    $aakoreInfo = $aakoreCmdOutput | ConvertFrom-Json
    return $aakoreInfo[2].server
}

function Get-AccessToken {
    param ([string] $cloudUrl, [string] $registrationToken)

    $headers = @{ "Content-Type" = "application/x-www-form-urlencoded" }
    $body = @{
        "grant_type" = "urn:ietf:params:oauth:grant-type:jwt-bearer"
        "assertion"  = $registrationToken
    }
    $response = Invoke-RestMethod -Uri "${cloudUrl}/bc/idp/token" `
        -Method Post `
        -Headers $headers `
        -Body $body

    return $response.access_token
}

function Invoke-Plan {
    param ([string] $cloudUrl, [string] $accessToken, [string] $resourceId)

    $headers = @{
        "Content-Type"  = "application/json"
        "Authorization" = "Bearer $accessToken"
    }
```

Name *: Acronis_manage_protection_plan

Description: Apply/Revoke existing Protection Plan

Category *: Acronis distribution kit Windows

Language *: PowerShell

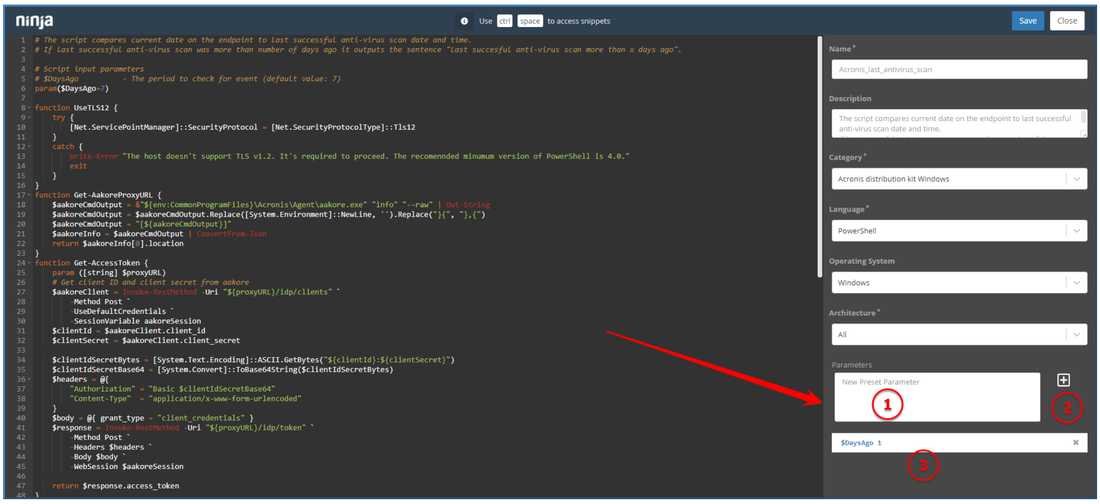
Operating System: Windows

Architecture *: All

Parameters: New Preset Parameter

- 4A84-4AE1-435A ×
- 4A84-4AE1-435A yes ×

iv. **Acronis_last_antivirus_scan** (Windows): add the $DaysAgo parameter, followed by a blank space and a number (the period to check for event - default value: 7)



```
# The script compares current date on the endpoint to last successful anti-virus scan date and time.
# If last successful anti-virus scan was more than number of days ago it outputs the sentence "Last succesful anti-virus scan more than x days ago".

# Script input parameters
# $DaysAgo          - The period to check for event (default value: 7)
param($DaysAgo=7)

function UseTLS12 {
    try {
        [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    }
    catch {
        Write-Error "The host doesn't support TLS v1.2. It's required to proceed. The recomennded minumum version of PowerShell is 4.0."
        exit
    }
}
function Get-AakoreProxyURL {
    $aakoreCmdOutput = &"${env:CommonProgramFiles}\Acronis\Agent\aakore.exe" "info" "--raw" | Out-String
    $aakoreCmdOutput = $aakoreCmdOutput.Replace([System.Environment]::NewLine, '').Replace("}{", "},{")
    $aakoreCmdOutput = "[${aakoreCmdOutput}]"
    $aakoreInfo = $aakoreCmdOutput | ConvertFrom-Json
    return $aakoreInfo[0].location
}
function Get-AccessToken {
    param ([string] $proxyURL)
    # Get client ID and client secret from aakore
    $aakoreClient = Invoke-RestMethod -Uri "${proxyURL}/idp/clients" `
        -Method Post `
        -UseDefaultCredentials `
        -SessionVariable aakoreSession
    $clientId = $aakoreClient.client_id
    $clientSecret = $aakoreClient.client_secret

    $clientIdSecretBytes = [System.Text.Encoding]::ASCII.GetBytes("${clientId}:${clientSecret}")
    $clientIdSecretBase64 = [System.Convert]::ToBase64String($clientIdSecretBytes)
    $headers = @{
        "Authorization" = "Basic $clientIdSecretBase64"
        "Content-Type"  = "application/x-www-form-urlencoded"
    }
    $body = @{ grant_type = "client_credentials" }
    $response = Invoke-RestMethod -Uri "${proxyURL}/idp/token" `
        -Method Post `
        -Headers $headers `
        -Body $body `
        -WebSession $aakoreSession

    return $response.access_token
}
```

Name *
Acronis_last_antivirus_scan

Description
The script compares current date on the endpoint to last successful anti-virus scan date and time.

Category *
Acronis distribution kit Windows

Language *
PowerShell

Operating System
Windows

Architecture *
All

Parameters
New Preset Parameter

$DaysAgo 1 ×

v. **Acronis_last_malware_scan** (Windows & macOS): add the $DaysAgo parameter, followed by a blank space and a number (the period to check for event - default value: 7)

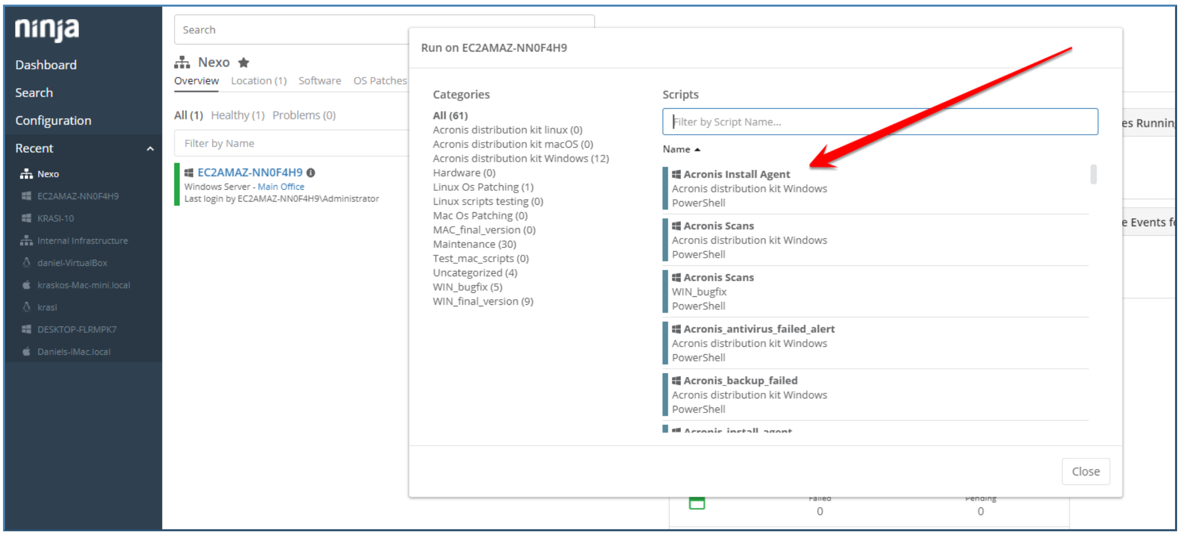
vi. The rest of the scripts do not require any preset parameters.

Note

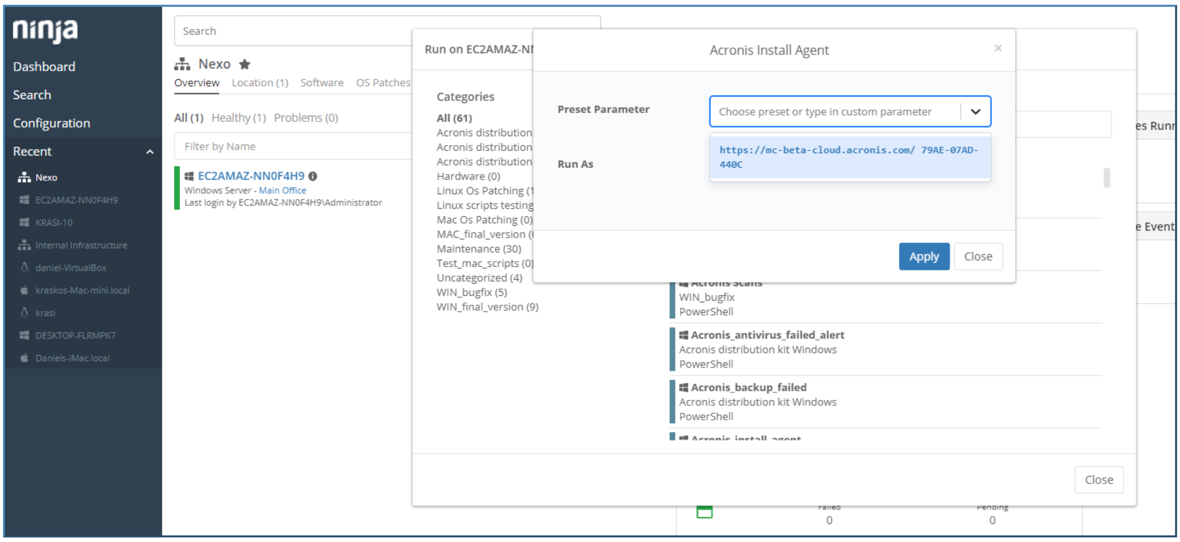
The NinjaRMMScriptsWindows.zip file contains a total of 10 scripts, NinjaRMMScriptsLinux.zip - 6 scripts and NinjaRMMScriptsMacOS.zip - 8.

7 Deploy Acronis Cyber Protection agent (Windows, Linux and macOS)

1. In the Ninja RMM interface, go to **Dashboard** > **Organizations** and click on the organization you want to manage.
2. Hover over a device of your choice, then navigate to **Run Script** and click **From Library**.
3. Click on the **Acronis Install Agent** script (pay attention to the device's OS and the script's target OS).



4. In the pop-up that appears next, click **Preset Parameter** and select the value configured for this script in the previous chapter.
5. In the **Run As** drop-down list, select the **System** value.


Acronis Install Agent

Preset Parameter: Choose preset or type in custom parameter

https://mc-beta-cloud.acronis.com/ 79AE-07AD-440C

Run As

6. Click **Apply**, then **Yes**.

8 Uninstall the Acronis Cyber Protection agent (Windows, Linux and macOS)

1. In the Ninja RMM interface, go to **Dashboard** > **Organizations** and click on the organization you want to manage.
2. Hover over a device of your choice, then navigate to **Run Script** and click **From Library**.
3. Click on the **Acronis Uninstall Agent** script (pay attention to the device's OS and the script's target OS).
4. In the pop-up that appears next:
   a. Leave the **Preset Parameter** field empty
   b. In the **Run As** drop-down list, select **System**.
5. Click **Apply**, then **Yes**.

9 Manage Protection Plan (Windows, Linux and macOS)

1. In the Ninja RMM interface, go to **Dashboard** > **Organizations** and click on the organization you want to manage.
2. Hover over a device of your choice, then navigate to **Run Script** and click **From Library**.
3. Click on the **Acronis manage protection plan** script (pay attention to the device's OS and the script's target OS).
4. In the pop-up that appears next:
   a. If you want to apply a protection plan, associated with this token: click **Preset Parameter** and select the value with **Acronis registration token** only
   b. If you want to revoke a protection plan, associated with this token, click **Preset Parameter** and select the value with **Acronis registration token**, followed by **yes**
5. In the **Run As** drop-down list, select the **System** value.
6. Click **Apply**, then **Yes**.

10 Perform Acronis scans and tasks (Windows, Linux and macOS)

1. In the Ninja RMM interface, go to **Dashboard** > **Organizations** and click on the organization you want to manage.
2. Hover over a device of your choice, then navigate to **Run Script** and click **From Library**.
3. Click on the **Acronis scans** script (pay attention to the device's OS and the script's target OS).
4. In the pop-up that appears next, click the **Preset Parameter** drop-down list and select one of the following values, which represents the type of task that needs to be run:
   - backup
   - av_scan
   - malware_scan
   - vulnerability_assessment
   - patch_management
5. In the **Run As** drop-down list, select the **System** value.
6. Click **Apply**, then **Yes**.

11 Monitoring

The following scripts are currently in use:

- **Acronis_backup_failed** - counts the number of open alerts of **backup failed** type and outputs the "**x backups failed**" sentence.

- **Acronis_antivirus_failed_alert** - counts the number of open alerts of "**Active Protection service is not running**" & "**Continuous Data Protection failed**" types and outputs the "**Active protection service failed x times**" sentence.

- **Acronis_malware_detected_alert** - counts cumulatively the total number of open alerts of "**Malware is detected and blocked (ODS)**" and "**Malware is detected and blocked (RTP)**" types and outputs the "**x MALWARE THREADS has been found**" sentence.

- **Acronis_last_backup** - compares the current endpoint date to the last successful backup date and time. If last successful backup is later than the number of days ago, it outputs the "**Last succesful backup more than x days ago**" sentence.

- **Acronis_last_antivirus_scan** - compares the current endpoint date to the last successful antivirus scan date and time. If last successful antivirus scan was later than the number of days ago, it outputs the "**Last succesful antivirus scan more than x days ago**" sentence.

- **Acronis_last_malware_scan** - compares the current endpoint date to the last successful antimalware scan date and time. If last successful antimalware scan was later than the number of days ago, it outputs the "**Last succesful antimalware scan more than x days ago"** sentence.

Each of the monitoring scripts can be set as either manually or repetitively executed with a Ninja RMM scheduled task.

In order to manually run a monitoring script:

1. In the Ninja RMM interface, go to **Dashboard** > **Organizations** and click on the organization you want to manage.
2. Hover over a device of your choice, then navigate to **Run Script** and click **From Library**.
3. Click on the monitoring script you want to run.
4. In the pop-up that appears next, select **Preset Parameter**, in case the script requires any.
5. In the **Run As** drop-down list, select the **System** value.
6. Click **Apply**, then **Yes**.

In order to create a Ninja RMM scheduled task for repetitive monitoring script execution:

1. Navigate to **Configuration** > **Tasks**.
2. Click **New Task** in the top right corner.
3. For the newly created task, specify:
    a. name
    b. schedule
    c. optionally, description.
4. Click **Add Script** in the top right to specify what script to run in the task. This will open the **Script Library**. You can select any of the Acronis monitoring scripts listed there.
5. While adding scripts to your scheduled task, you will be prompted to specify **Preset parameters** and one of the following ways to run the script as:
    - System
    - current user
    - using your store credentials.
6. In the **Run As** drop-down list, select the **System** value.
7. Next, navigate to the **Targets** tab on the left page side.
8. Click **Add** in the top right.
9. Specify any desired organization(s), device(s) and/or group(s) to run the task on, then click **Apply**.
10. Finally, click **Save**.