



Creating and Deploying a Provider-specific IPAM Integration Package for VMware Aria Automation

This document applies to vendor-specific IPAM integration in VMware Aria Automation using the available IPAM SDK.

TECHNICAL PAPER

APRIL 2023

VERSION 1.9

Table of Contents

Summary.....	3
Running environment.....	4
IPAM operation definitions.....	6
Baseline contract between Automation Assembler IPAM service and external IPAM providers	9
External IPAM packaging format.....	25
Contents of a sample IPAM .zip package.....	27
Download IPAM SDK and implement external IPAM integration.....	28
Considerations and tips.....	39
Learn more – Videos and Documentation.....	40
Appendix A – Format for endpoint-schema.json	40



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Revision History

DATE	VERSION	DESCRIPTION
October 17, 2019	1.0	Initial version.
December 6, 2019	1.1	Title change and note about forthcoming SDK.
April 14, 2020	1.2	Includes SDK and documentation updates for IPAM SDK 1.0.0.
June 07, 2020	1.3	Column title correction from Optional to Required in Baseline contract.
June 20, 2020	1.4	Add link to IPAM SDK video and blog post.
September 9, 2020	1.5	Language cleanup.
October 6, 2020	1.6	Add IPAM SDK 1.1.0 links and information.
January 21, 2021	1.7	Update Entity: information for ResourceInfo, AddressInfo, IpDeallocation, IpRangeAllocation, IpRangeDeallocation.
September 15, 2021	1.8	Update.
April 8, 2023	1.9	Branding updates for VMware Aria Automation.

Summary

The goal of this document is to provide the information needed by external IPAM providers to integrate their external IPAM system with the Automation Assembler service in VMware Aria Automation.

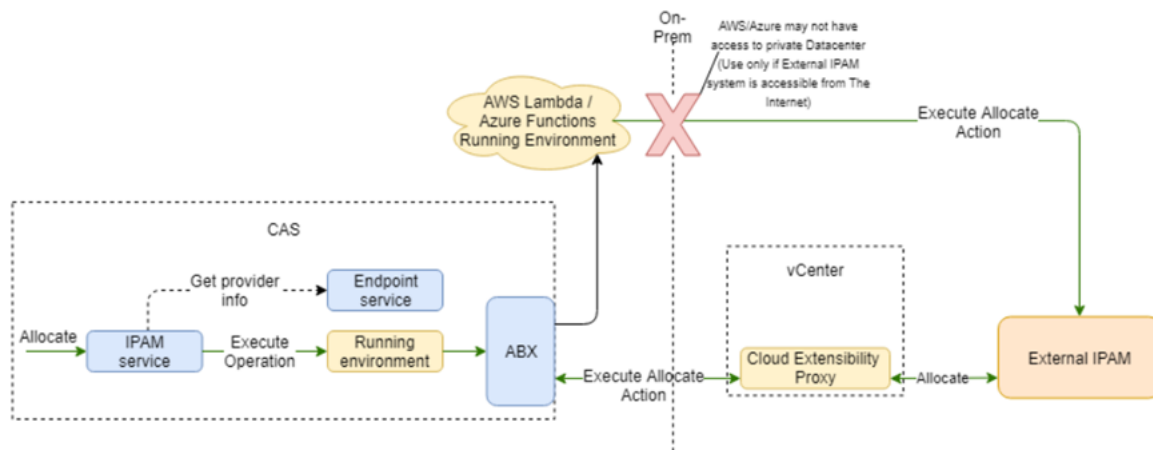
Reference this document when building a custom external IPAM integration for VMware Aria Automation.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler



After you create the external IPAM integration package using the instruction provided in this document, you can use the following workflow scenarios in the Automation Assembler product documentation to create and use the IPAM integration point.

Note: New IPAM SDK integration packages are released periodically. The format of the IPAM SDK integration package may change based on the introduction of new SDK packages. To support backward compatibility, the existing format continues to be supported but use of the updated format is preferred.

Running environment

The running environment is the communication engine between Automation Assembler and the external IPAM system. Integrators of external IPAM systems work with the tools provided by the running environment to build a set of scripts and workflows that can execute IPAM operations.

You implement one script or workflow for each operation that the IPAM service supports. The IPAM service sends requests in a properly defined format to the running environment and asks it to perform a certain IPAM operation, such as **Allocate IP for VM** or **Obtain a list of IP ranges**.

To complete the IPAM operation, the running environment executes the respective script or workflow that performs that specific task.

Currently, the only supported running environment is actions-based extensibility or ABX. You create ABX workflows in Automation Assembler in either VMware Aria Automation.

With ABX, you can use the full potential of FaaS services such as AWS Lambda, Azure Functions, and OpenFaaS (action-based extensibility on-prem). Author the source code scripts in Python, NodeJS, or any other language that ABX supports.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

For more information about creating a running environment within the context of a sample external IPAM integration workflow, see the following product documentation:

- VMware Aria Automation (SaaS) – [Create a running environment for an IPAM integration point](#)
- VMware Aria Automation (on-prem) – [Create a running environment for an IPAM integration point](#)



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

IPAM operation definitions

Automation Assembler in VMware Aria Automation supports these IPAM service operations. Operation inputs are received as script function or method arguments.

Operation name	Description	Input	Output	Required
Allocate IP	Allocate next available IP address for each Allocation info.	IpAllocationRequest	IpAllocationResponse	Yes
Deallocate IP	Release allocated IP addresses.	IpDeallocationRequest	IpDeallocationResponse	Yes
Get IP Ranges	Get a page of IP ranges from IPAM endpoint.	GetIpRangesRequest	GetIpRangesResponse	Yes
Update Record	Updates the created host record. Could be used to update MAC address of machine after it has been provisioned.	RecordUpdateRequest	RecordUpdateRequest	No
Validate Endpoint	Validates that the IPAM endpoint	EndpointValidationRequest	EndpointValidationResponse	Yes



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Operation name	Description	Input	Output	Required
	credentials are valid and that a connection to the external IPAM system can be established successfully.			
Allocate IP Range Note: Unavailable for vRealize Automation 8.0.x.	Creates a network inside an IP block.	IpRangeAllocationRequest	IpRangeAllocationResponse	No
Deallocate IP Range Note: Unavailable for vRealize Automation 8.0.x.	Deletes an already allocated network.	IpRangeDeallocationRequest	IpRangeDeallocationResponse	No
Get IP Blocks Note: Unavailable for vRealize Automation 8.0.x.	Get page of IP blocks from IPAM endpoint.	EnumerationRequestBase	GetIpBlocksResponse	No



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Baseline contract between Automation Assembler IPAM service and external IPAM providers

Automation Assembler in VMware Aria Automation supports these IPAM service baseline contracts for the integrated external IPAM provider.

Entity: ProviderRequestBase

Property	Type	Required	Description
endpoint	Endpoint	Yes	A parent class for all request types DTOs. Provides basic information about the IPAM provider endpoint in Automation Assembler.

Entity: ProviderResponseBase

Note: This entity has been deprecated.

Property	Type	Required	Description
error	ErrorStatus	No	DEPRECATED This property was initially used to propagate errors from the plug-in to VMware Aria Automation. This field is deprecated and replaced by exceptions. In case of an error, the plug-in is expected to use the capabilities of the underlying running environment to report that error. For an ABX running environment, an exception is thrown inside the action.

Entity: ErrorStatus

Note: This entity has been deprecated.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
errorCode	Integer	Yes	DEPRECATED This property was initially used to propagate errors from the plug-in to VMware Aria Automation. This field is deprecated and replaced by exceptions. In case of an error, the plug-in is expected to use the capabilities of the underlying running environment to report that error. For an ABX running environment, an exception is thrown inside the action.
errorMessage	String	Yes	DEPRECATED

Entity: EndpointValidationRequest

Property	Type	Required	Description
authCredentialsLink	String	Yes	Carries host and credentials data about the external IPAM provider that is needed to validate the connection. Contains the link to the credentials store where the external IPAM provider credentials are kept in VMware Aria Automation.
endpointProperties	Map<String, String>	Yes	Carries host and credentials data about the external IPAM provider that is needed to validate the connection. Contains a collection that holds provider-specific endpoint properties such as hostname and others that are defined in the endpoint-schema.json file.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Entity: EndpointValidationResponse: ProviderResponseBase

Property	Type	Required	Description
message	String	No	Response of the EndpointValidationRequest. Indicates successful validation of endpoint credentials and connectivity.
certificateInfo	CertificateInfo	No	Response of the EndpointValidationRequest. If the certificate for the external IPAM provider is not automatically trusted, this field can be used with the error message to propagate the certificate to Automation Assembler user and allow the user to manually confirm certificate trust.

Entity: Endpoint

Property	Type	Required	Description
id	String	Yes	An entity representing the registered external IPAM provider, for example Infoblox. This is a VMware Aria Automation endpoint ID
authCredentialsLink	String	Yes	An entity representing the registered external IPAM provider, for example Infoblox. Contains a link to an AuthCredentialsState document in the VMware Aria Automation database that stores credentials in a secure way. Can be accessed from an ABX script by using <i>context.request(<link>, GET)</i> syntax.
endpointProperties	String	Yes	An entity that represents the registered external IPAM provider, for example Infoblox.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
			Contains IPAM provider endpoint properties, such as hostname, that are defined in the endpoint-schema.json file.

Entity: PagingAndSorting

Property	Type	Required	Description
maxResults	Integer	No	Specifies the maximum number of returned results per page. If the number of available results is larger than maxResults, the IPAM provider must return a nextPageToken to get the next page of results in subsequent list requests.
pageToken	String	No	Specifies a page token to use. To get the next page of results, set the pageToken to the nextPageToken returned by a previous list request.

Entity: EnumerationRequestBase

Property	Type	Required	Description
Endpoint	Endpoint	Yes	Contains request data required to enumerate any kind of resources. Contains basic information about the IPAM integration in VMware Aria Automation.
pagingAndSorting	PagingAndSorting	Yes	Contains request data required to enumerate any kind of resources. Contains a key-value entry containing pagination related data.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Entity: GetIpRangesResponse : ProviderResponseBase

Property	Type	Required	Description
ipRanges	List<IpRange>	Yes	Contains the list of IP ranges returned in the result set.
nextPageToken	String	No	Allows you to get the next page of results for list requests. If the number of results is larger than maxResults (see above entity), use the nextPageToken as a value for the query parameter pageToken in the next list request. Subsequent list requests have their own nextPageToken to continue paging through the results.

Entity: IpRange

Property	Type	Required	Description
id	String	Yes	Provider specific ID. If the provider doesn't have IDs, the workflow/action can generate an ID using a combination of range start address, end address, and address space.
name	String	Yes	User friendly name.
description	String	No	User friendly description.
startIPAddress	String	Yes	Range start IP address.
endIPAddress	String	Yes	Range end IP address.
ipVersion	Enum: {IPv4, IPv6}	Yes	Range IP version – IPv4 or IPv6.
addressSpaceId	String	No	Address space where the range belongs.
gatewayAddress	String	No	The gateway IP address.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
subnetPrefixLength	Integer	Yes	The length of the subnet mask
dnsServerAddresses	List<String>	No	Ordered list of DNS servers.
domain	String	No	DNS domain of this range.
dnsSearchDomains	List<String>	No	Ordered list of DNS domain search.
properties	Map<String, String>	No	Provider specific range properties.
tags	List<Tag>	No	Tags collection, for example the extensible attributes in Infoblox.

Entity: IpAllocationRequest : ProviderRequestBase

Property	Type	Required	Description
resourceInfo	ResourceInfo	Yes	Contains information about the resource, for example a machine or load balancer, to which the range is associated.
ipAllocations	List<IpAllocation>	Yes	Contains the list of allocations to be reserved for this resource. For example, a machine may have multiple NICs and a separate IP allocation should be supplied for each NIC.

Entity: ResourceInfo

Property	Type	Required	Description
id	String	Yes	The resource ID, for example PhM documentSelfLink.
name	String	Yes	The resource name.
hostName	String	No	The resource hostname.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
description	String	No	The resource description.
type	Enum {VM, LOADBALANCER, NAT, OTHER}	Yes	The resource type.
owner	String	Yes	The resource owner, for example fritza@vmware.com).
orgId	String	Yes	The tenant ID or organization ID in which the resource resides.
properties	Map<String, String>	No	The resource-specific custom properties.
projectId Note: Not available for vRealize Automation 8.2 or earlier.	String	No	The ID of the project this entity belongs to.
templateId Note: Not available for vRealize Automation 8.2 or earlier.	String	No	The ID of the template/cloud template this entity belongs to.
deploymentId Note: Not available for vRealize Automation 8.2 or earlier.	String	No	The ID of the deployment this entity belongs to.

Entity: IpAllocation



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
id	String	Yes	ID of the IpAllocation. This ID is set by the IPAM service to match the IpAllocation value with the corresponding AllocationResult value.
description	String	No	Description for allocation request in human-readable format, for example in CIDR format.
ipRangeIds	List<String>	Yes	Provider-specific range IDs to use to allocate IP addresses. To avoid DNS and gateway information mismatch, only one range is used to allocate all IP addresses.
nicIndex	Integer	Yes	Nic index of the resource.
isPrimary	Boolean	Yes	Set to true when allocating a primary IP for the Nic Index of the resource.
start	String	No	Start IP address. Used to allocate specific IP address.
size	Integer	Yes	Number of IP addresses to allocate. Used to allocate more than one IP address. If <i>Start</i> is also specified, continues IP addresses to be allocated.
properties	Map<String, String>	Yes	May contain the provider-specific properties.

Entity: IpAllocationResponse : ProviderResponseBase

Property	Type	Required	Description
ipAllocations	List<AllocationResult>	Yes	List of IP allocations to match with allocations in the request.

Entity: AllocationResult



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
ipAllocationId	String	Yes	ID of the specified IpAllocation. This is set by the IPAM service to match the IpAllocation value with the corresponding AllocationResult value.
ipAddresses	List<String>	Yes	Allocated IP addresses. If a start address was specified, this is a continuous block of IP addresses. If no start address was specified, any IP addresses within a single range are allocated.
ipRangeId	String	Yes	Provide-specific range ID used to allocate IP addresses.
ipVersion	Enum {IPv4, IPv6}	Yes	IP version range.
gatewayAddresses	List<String>	No	Gateway addresses. Can be used to overwrite the value from the IP range
subnetPrefixLength	Integer	No	Subnet prefix length (synonymous with netmask). Can be used to overwrite the value from the IP range.
dnsServerAddresses	List<String>	No	DNS IP addresses. Can be used to overwrite the value from the IP range.
Domain	String	No	DNS domain. Can be used to overwrite the value from the IP range.
dnsSearchDomains	List<String>	No	DNS domain search, in order. Can be used to overwrite the value from the IP range.
Properties	Map<String, String>	No	Provider-specific allocation result properties that the provider needs to be set on the resource NIC as machine NIC custom properties.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Entity: IpDeallocationRequest : ProviderRequestBase

Property	Type	Required	Description
resourceInfo	ResourceInfo	Yes	Contains information about the resource, for example a machine or load balancer, that the range is associated with.
ipDeallocations	List<IpDeallocation>	Yes	List of deallocations containing information about which IPs to release for the resource.

Entity: IpDeallocation

Property	Type	Required	Description
id	String	Yes	ID of IpDeallocation. The ID is set by the IPAM service to match the IpDeallocation value with the corresponding DeallocationResult value.
ipAddress	String	Yes	The IP address to deallocate.
ipRangeId	String	Yes	Provider-specific range ID of the IP range used for allocating this IP address.
vraIpAddressId Note: Not available for vRealize Automation 8.2 or earlier.	String	Yes	The ID of the IP address instance in VMware Aria Automation.
addressSpaceId Note: Not available for vRealize Automation 8.2 or earlier.	String	No	The ID of the address space that the range belongs to.

Entity: IpDeallocationResponse : ProviderResponseBase



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
ipDeallocations	List<DeallocationResult>	Yes	List of deallocations to match with deallocation IPs from the request.

Entity: DeallocationResult

Property	Type	Required	Description
ipDeallocationId	String	Yes	ID of IpDeallocation. The ID is set by the IPAM service to match the IpDeallocation value with the corresponding DeallocationResult value.
message	String	No	Optional message to include in the deallocation response.

Entity: RecordUpdateRequest : ProviderRequestBase

Property	Type	Required	Description
resourceInfo	ResourceInfo	Yes	Information about the resource, for example the machine or load balancer, to which this IP range is associated.
hostName	String	Yes	The resource hostname.
addressInfos	List<AddressInfo>	Yes	List of address information to be used for updating the MAC address of the record.

Entity: AddressInfo

Property	Type	Required	Description
address	String	Yes	IP address of the record.
macAddress	String	Yes	MAC address of the record.
nicIndex	Integer	Yes	0-based index of the NIC.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
ipAllocationId Note: Not available for vRealize Automation 8.2 or earlier.	String	Yes	The ID of the allocation. This ID is set by the IPAM service to match UpdateRecord with the corresponding allocation.
vralPAddressId Note: Not available for vRealize Automation 8.2 or earlier.	String	Yes	The ID of the IP address instance in VMware Aria Automation that was created during allocation.
addressSpaceId Note: Not available for vRealize Automation 8.2 or earlier.	String	No	The ID of the address space that the range belongs to.

Entity: IpRangeAllocation – Holds data required to allocate a network IP range

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
name	String	Yes	Network name.
description	String	No	Network description.
ipBlockIds	List<String>	Yes	List of IP blocks that can be used to allocate the network. The List type allows you to specify multiple blocks, for example to allocate the network inside the first block that matches a requirement.
addressSpaceId	String	No	Address space where the range belongs.
gatewayAddress	String	No	Gateway IP address.
subnetCidr	String	No	CIDR that can be used to allocate the network on a specific IP address.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
subnetPrefixLength	Integer	Yes	Subnet mask length.
dnsServerAddresses	List<String>	No	DNS IP addresses for this network.
domain	String	No	DNS domain of this network.
dnsSearchDomains	List<String>	No	Ordered list of DNS domain search.
properties	Map<String, String>	No	Provider specific properties.
tags	List<Tag>	No	Tags collection, for example the extensible attributes in Infoblox.
id Note: Not available for vRealize Automation 8.2 or earlier.	String	Yes	The ID of the IP range allocation. This ID is set by the IPAM service to match IpRangeAllocation with the corresponding deallocation.

Entity: IpRangeAllocationRequest : ProviderRequestBase

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
resourceInfo	ResourceInfo	Yes	Information about the resource, for example the machine or load balancer, to which this IP range is associated.
ipRangeAllocation	IpRangeAllocation	Yes	Information about the IP range to be allocated.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Entity: IpRangeAllocationResponse : ProviderResponseBase

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
ipRange	IpRange	Yes	The state of the network's allocated IP range.

Entity: IpRangeDeallocation – Holds data required to deallocate a network IP range

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
ipRangeId	String	Yes	The ID of the network to be deallocated.
addressSpaceId	String	No	The address space that the IP range belongs to.
properties	Map<String, String>	No	Provider-specific properties.
id Note: Not available for vRealize Automation 8.2 or earlier.	String	Yes	The ID of the IP range deallocation. This ID is set by the IPAM service to match IpRangeDeallocation with the corresponding allocation.

Entity: IpRangeDeallocationRequest: ProviderRequestBase

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
resourceInfo	ResourceInfo	Yes	Information about the resource, for example the machine or load balancer, to which this IP range is associated.
ipRangeDeallocation	IpRangeDeallocation	Yes	Information about the IP range to be deallocated.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Entity: IpRangeDeallocationResponse

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
message	String	No	A message that indicates whether the IP range was successfully deallocated.

Entity: IpBlock – Holds data associated with a single IP Block

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
id	String	Yes	Provider-specific ID. If the provider does not provide an ID, the action or workflow can generate one by combining the block name and the CIDR.
name	String	Yes	IP block name. For example, the IP block name might be the block CIDR.
description	String	No	IP block description.
ipBlockCIDR	String	Yes	The IP block CIDR, for example 192.168.0.0/24.
ipVersion	enum: {IPv4, IPv6}	Yes	Block type - either IPv4 or IPv6.
addressSpace	String	No	Address space where the block belongs.
gatewayAddress	String	No	Gateway IP address, for example 192.168.0.1.
dnsServerAddresses	List<String>	No	DNS IP addresses for this block.
domain	String	No	DNS domain of this block.
dnsSearchDomains	List<String>	No	Ordered list of DNS domains.
properties	Map<String, String>	No	Provider specific properties.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Property	Type	Required	Description
tags	List<Tag>	No	Tags collection, for example the extensible attributes in Infoblox.

Entity: GetIpBlocksResponse

Note: Unavailable for vRealize Automation 8.0.x.

Property	Type	Required	Description
ipBlocks	List<IpBlock>	Yes	Contains information about multiple IP blocks.

External IPAM packaging format

IPAM operations (in the form of ABX scripts) are packaged with configurations and metadata into an IPAM .zip file. You can create a vendor-specific IPAM package by using the supplied *VMware Aria Automation Third-Party IPAM SDK* available at [VMware code](#) as described later in this document.

The IPAM .zip file is uploaded to [VMware Marketplace \(https://marketplace.cloud.vmware.com/\)](https://marketplace.cloud.vmware.com/) or to the integrator's own web site customer for customer download and deployment to VMware Aria Automation. After deployment, the external IPAM integration is visible by using the Automation Assembler menu sequence **Infrastructure** → **Connections** → **Integrations** → **Add Integration** → **IPAM** → **Provider**.

New Integration

IPAM Provider
Specify IPAM provider for integration. Add new if no providers are registered in Cloud Assembly

Provider *

[IMPORT PROVIDER PACKAGE](#)

[ADD](#) [CANCEL](#)



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.




Contents of a sample IPAM .zip package

You can create a vendor-specific IPAM .zip package to support an external IPAM provider in VMware Aria Automation by using the supplied *VMware Aria Automation Third-Party IPAM SDK* available at [VMware code](#):

- [VMware Aria Automation Third-Party IPAM SDK 1.1.0](#)
- [VMware Aria Automation Third-Party IPAM SDK 1.0.0](#)

The IPAM package is written in python and uses an ABX running environment. The .zip file contents include custom implementations of the currently supported IPAM operations, including - **Allocate IP**, **Deallocate IP**, **Get IP Ranges**, **Update Record**, and **Validate Endpoint**.

An IPAM .zip package consists of the following files: **bundle.zip**, **endpoint-schema.json**, **logo.png**, and **registration.yaml**.

Name	Type	Compressed size	Password ...	Size	Ratio
 bundle.zip	Compressed (zipped) Fol...	2 KB	No	2 KB	33%
 endpoint-schema.json	JSON File	1 KB	No	1 KB	62%
 registration.yaml	YAML File	1 KB	No	1 KB	45%

- **registration.yaml:**
Format (can be extended in the future):

```

---
name: "Sample IPAM"
description: "Sample IPAM integration for CAS"
version: "0.1"

abxConfig:
  allocateIPActionId: "SampleIPAM_AllocateIP"
  deallocateIPActionId: "SampleIPAM_DeallocateIP"
  validateEndpointActionId: "SampleIPAM_ValidateEndpoint"
  updateResourceActionId: "SampleIPAM_Update"
  getIPRangesActionId: "SampleIPAM_GetIPRanges"

properties:
  supportsUpdateRecord: true
  supportsOnDemandNetworks: false

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Description: Contains meta information about the contents of the IPAM .zip. Describes the ABX action IDs to invoke for the different types of IPAM operations. The IPAM provider name and description is also stored here. The name, description, and properties of the IPAM provider are also stored here.

- **endpoint-schema.json**

Format: The form definition format is documented in **Appendix A**.

Description: Contains the custom form definition that renders the IPAM provider's specific fields during IPAM endpoint registration.

Important: The **endpoint-schema.json** file must contain entries for **privateKey** and **privateKeyId** fields.

These fields indicate sensitive data within the custom form that must be stored in a secure way.

- **bundle.zip**

Format: Uses the same format as ABX for exporting sets of actions.

Description: Contains the set of ABX actions in the established ABX format.

- **logo.png**

Description: Contains the logo icon for the specific IPAM provider.

Download IPAM SDK and implement external IPAM integration

To create an IPAM .zip package for use with your specific provider, download and use the external IPAM SDK **vra-third-party-ipam-sdk** as described in the following process.

The IPAM SDK package is available from [VMware code](#) as follows:

- [VMware Aria Automation Third-Party IPAM SDK 1.1.0](#)
- [VMware Aria Automation Third-Party IPAM SDK 1.0.0](#)

Note: The related procedure for creating ABX actions in Automation Assembler is described in the following product documentation topics:

- VMware Aria Automation (SaaS) – [Learn more about extensibility actions](#)
- VMware Aria Automation (on-prem) – [Learn more about extensibility actions](#)



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Prerequisites

To use the external IPAM SDK, you must install and configure the following software. You can use a higher version of each, provided that the version is backward compatible.

- Java 8
- Maven 3 (Used for packaging the IPAM zip.)
- Python 3 (The IPAM plug-in is based on Python.)
- Docker (Docker is used to collect the Python dependency libraries needed by the IPAM plug-in.)
- Internet access (The IPAM SDK relies on Maven Central, Docker HUB & PIP during packaging time. Internet access is not required at runtime.)



Step 1 – Download the IPAM Integration SDK from VMware Code

Download the most recent *VMware Aria Automation Third-Party IPAM SDK* from the [VMware code](#) site:

- [VMware Aria Automation Third-Party IPAM SDK 1.1.0](#)
- [VMware Aria Automation Third-Party IPAM SDK 1.0.0](#)

A README file with needed instructions is supplied with the IPAM SDK download. The README file content is summarized in the following steps.

Step 2 – Package the scripts

Maven and Docker are used during build time to package the Python scripts into an IPAM .zip distribution. Maven enables the building of the IPAM package to be platform independent. This allows integrators to develop their IPAM integration solution under any Java-enabled operating system. Docker is used during build time to start up a Photon OS container. All 3rd party libraries that the IPAM plugin depends on are downloaded during build time, using PIP, from within the Photon operating system Docker container. This guarantees that all Python library binaries are compiled correctly for the Photon operating system, which is the operating system of the Running Environment that executes the IPAM Python actions.

1. Open the **pom.xml**, which resides in the **root** directory, and modify the following properties:

```
<provider.name>SampleIPAM</provider.name> <provider.description>Sample  
IPAM integration for vRA</provider.description>  
<provider.version>0.1</provider.version>
```

Replace the property values with the name, description, and version of your choice. The **provider.name** is used as a display name in VMware Aria Automation when you deploy the plugin zip, along with the description and version.

2. Update the **logo.png** file with the logo icon of your company.

VMware Aria Automation uses the **logo.png** file located in the **./src/main/resources** when displaying the IPAM endpoints that you create by using this package.

3. (Optional) Change the IPAM Integration endpoint custom form.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Do this by modifying the **endpoint-schema.json** file in the **./src/main/resources** folder. This **.json** file contains the custom form definition that renders the IPAM provider's specific fields during IPAM endpoint registration. You can change the form, but the file must contain entries for the **privateKey** and **privateKeyId** fields.

Note: The **registration.yaml** file also resides in the **./src/main/resources** folder. It contains meta information about the contents of the package. Do not change anything in the **registration.yaml** file.

4. From the **root** directory, run the following command:

```
run mvn package -PcollectDependencies
```

This produces a **SampleIPAM-with-dependencies.zip** file under the **./target** folder. The zip file is ready to be deployed into VMware Aria Automation.

The first time that you run this command, it can take several minutes to complete packaging the IPAM zip file. The first time the script runs, it attempts to collect any required 3rd party Python libraries, such as **requests** and **pyopssh**.

Subsequent runs of the `mvn package` command do not trigger another collection of 3rd party libraries. To re-trigger the collection of these dependencies, you must provide the `-PcollectDependencies` option in the command line.

The **SampleIPAM-with-dependencies.zip** IPAM package is now ready to use.

You can test the IPAM package by uploading it in VMware Aria Automation and create an IPAM integration. Check that expected actions are triggered and are executing successfully. For example, create a new IPAM endpoint and choose the package you uploaded in the **Provider** dropdown, enter an arbitrary username and password, enter **httpbin.org** as a **Hostname** and click on **Validate**. You should see the **Validate Endpoint** action is triggered in the **Extensibility** tab. It should complete successfully.



Step 3 – Get familiar with the IPAM operations and their skeleton implementations

After checking that the packaging of the sample IPAM scripts works, you can start exploring the code. In the `./src/main/python` folder, there is a separate directory for each IPAM-specific operation that the plug-in supports.

Operation name	Description	Script	Required
Allocate IP	Allocates the next available IP for a VM.	<code>./src/main/python/allocate_ip/source.py</code>	Yes
Deallocate IP	Deallocates an already allocated IP.	<code>./src/main/python/deallocate_ip/source.py</code>	Yes
Get IP Ranges	Data collects IP ranges & networks from the IPAM provider.	<code>./src/main/python/get_ip_ranges/source.py</code>	Yes
Update Record	Updates the created host record. Could be used to update MAC address of VM after it has been provisioned.	<code>./src/main/python/update_record/source.py</code>	No
Validate Endpoint	Validates that the IPAM endpoint credentials are valid and that a connection to the external	<code>./src/main/python/validate_endpoint/source.py</code>	Yes



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Operation name	Description	Script	Required
	IPAM system can be established successfully.		
Allocate IP Range	Creates network inside some of the specified IP blocks.	<code>./src/main/python/allocate_ip_range/source.py</code>	No
Deallocate IP Range	Deletes an already allocated network.	<code>./src/main/python/deallocate_ip_range/source.py</code>	No
Get IP Blocks	Data collects IP blocks	<code>./src/main/python/get_ip_blocks/source.py</code>	No

The `./src/main/python/**/source.py` scripts contain the Python source code that would be used by VMware Aria Automation to perform the respective IPAM operation.

Each script defines a `def handler(context, inputs) :` function that is the entry point into the IPAM operation.

The VMware Aria Automation IPAM framework calls the respective operation's `handler` function, passing request specific inputs in the form of a Python dictionary. The request also includes a context object that can be used to securely connect to VMware Aria Automation and call its services.

Step 4: Implement the IPAM operations

You can implement the `def handler(context, inputs) :` function of each IPAM operation's `source.py` script but you must adhere to the contract defined in the **Baseline contract between Automation Assembler IPAM service and external IPAM providers** section of this document.

Implementing the operations from scratch is not advised. Instead, use the `vra_ipam_utils` library located in `./src/main/python/commons/vra_ipam_utils`.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

This library contains utility functions and classes to help with your `def handler(context, inputs):` implementation.

The `source.py` code uses the `vra_ipam_utils` library, so you can refer to it as reference:

```
def handler(context, inputs):
    ipam = IPAM(context, inputs)
    IPAM.do_validate_endpoint = do_validate_endpoint

    return ipam.validate_endpoint()

def do_validate_endpoint(self, auth_credentials, cert):
    # Your implementation goes here
    ...
```

To implement an operation, add your specific logic in the places indicated by the comments in the corresponding `source.py` file.

Tip: Build the package, upload it in VMware Aria Automation, and test it after implementing each operation.

Implement the IPAM operations sequentially in the following order:

1. Validate Endpoint
2. Get IP Ranges
3. Get IP Blocks (Optional)
4. Allocate IP
5. Allocate IP Range (Optional)
6. Deallocate IP
7. Deallocate IP Range (Optional)
8. Update Record (Optional)

You can execute REST calls against in VMware Aria Automation from within the Python scripts by using the `context` object in your handler:

```
context.request(link='/iaas/api/machines', operation='GET', body='')
```

The `context` is configured to handle request authentication, authorization, and proxy needs.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

The GetIPRanges and GetIPBlocks actions can only be run by a provisioning service user. All other actions can be run by the user who deployed the cloud template or catalog item.

Note: Requests that are run by the context object are initiated on behalf of the user who started the action event in VMware Aria Automation. When using `/iaas/api/machines` calls, there are two options available:

Option 1 - The provisioning user starts an enumeration action such as GetIPRanges or GetIPBlocks.

Option 2 - The user that initiated the provisioning request (for example, the administrator or Automation Consumption user or the Automation Assembler user who deployed the cloud template or catalog item) starts a validate, allocate, deallocate, or update action.

User rights information for the action request appears in the `context.request` within the `Started by` (or `triggeredBy`) field. API requests are validated based on user rights. For example, an action request run by a Automation Consumption user who lacks admin rights fails because the API requires the user to have admin rights. But the provisioning user, or admin user, who has admin rights can successfully run the action request because they have the required admin rights.

Step 5: Define 3rd party libraries (if needed)

To use 3rd party Python libraries in the `source.py` scripts, define them in the `requirements.txt` file located next to each IPAM operation's `source.py` script.

The plug-in build script downloads the dependency libraries that are defined in the `requirements.txt` file and package them in the correct format within the IPAM `.zip` file.

Always re-run the `mvn package -PcollectDependencies` command every time you add or remove a new dependency from the `requirements.txt` file.

The `requirements.txt` format is defined at <https://pip.readthedocs.io/en/1.1/requirements.html>.

Step 6: Change specific properties in the pom.xml file (if needed)

There are several optional operations.

A - Implement the optional Update Record operation

You can implement the *Update Record* operation. This operation is used by the IPAM service to notify



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

the external IPAM system that a VM has been successfully provisioned. It is also used to propagate the VM's MAC address to the IPAM system.

Support of this optional operation is controlled by the following property in the **pom.xml** file:

```
<provider.supportsUpdateRecord>true</provider.supportsUpdateRecord>
```

Changing this value to false excludes update operation from the IPAM .zip package.

Note: If you change the setting from false to true, you must re-run the `mvn package -PcollectDependencies` command to collect the required dependencies.

B - Implement the optional Get IP Blocks, Allocate IP Range, and Deallocate IP Range operations

These three operations are part of the extended IPAM plugin specification for VMware Aria Automation. They enable the plug-in to support provisioning of on-demand networks from VMware Aria Automation.

When a VMware Aria Automation user requests provisioning for an on-demand network, a CIDR for that network is allocated from the plug-in along, along with other network settings such as default gateway.

Support for these operations is controlled by the following property in the **pom.xml** file:

```
<provider.supportsOnDemandNetworks>false</provider.supportsOnDemandNetworks>
```

Changing the setting to true forces the build to include the `get_ip_blocks`, `allocate_ip_range`, and `deallocate_ip_range` operations inside the IPAM zip package.

Note: If you change the setting from false to true, you must re-run the `mvn package -PcollectDependencies` command to collect the required dependencies.

C - Support address spaces

External IPAM networks and ranges can be organized into logical groups with overlapping address spaces, serving a single routing domain. By default, the sample IPAM .zip that this SDK produces is configured to not support address spaces. If your IPAM system supports address spaces, you can enable support for address spaces by changing the following property in the **pom.xml** file:

```
<provider.supportsAddressSpaces>true</provider.supportsAddressSpaces>
```

Step 7: Build the package with the implemented IPAM



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

It is a good idea to deploy the package to VMware Aria Automation and test the operations after implementing each IPAM operation. Build the package by running `mvn package` or `mvn package -PcollectDependencies`. After you implement and test all the operations, the IPAM package is ready to be distributed and used.

Troubleshooting

The following list contains the most common errors that might occur during build time:

1. The `mvn package` build fails with the following message:

```
[ERROR] Plugin org.apache.maven.plugins:maven-resources-plugin:3.1.0 or
one of its dependencies could not be resolved: Failed to read artifact
descriptor for org.apache.maven.plugins:maven-resources-
plugin:jar:3.1.0: Could not transfer artifact
org.apache.maven.plugins:maven-resources-plugin:pom:3.1.0 from/to
central (https://repo.maven.apache.org/maven2): repo.maven.apache.org:
Unknown host repo.maven.apache.org -> [Help 1]
```

Resolution: The connection to Maven Central may have timed out or failed. Retry after a couple of minutes. If the issue persists, check your internet connection.

2. The `mvn package -PcollectDependencies` build fail with the following message:

```
[ERROR] DOCKER> Unable to pull 'vmware/photon2:20180424' : error
pulling image configuration: Get
https://production.cloudflare.docker.com/registry-
v2/docker/registry/v2/blobs/sha256/12/1204ad97f071063bea855f351348e15e9
cc03610cbfc8df46ab96b42d7cafa9f/data?verify=1578042999-
Nu9yKJgKQcuFU0Y9hAQe%2BKEOKGo%3D: dial tcp: lookup
production.cloudflare.docker.com on XXX:53: read udp XXX:57798->XXX:53:
i/o timeout
```

Resolution: The connection to the Docker Registry may have timed out or failed. Retry after a couple of minutes. If the issue persists, check your internet connection.

3. The `mvn package -PcollectDependencies` build on Windows fails with the following message:

```
[ERROR] Failed to execute goal io.fabric8:docker-maven-
plugin:0.31.0:start (start-container) on project sample-ipam: I/O
Error: Unable to create container for [ipam-dependency-
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
collector:latest] : Drive has not been shared (Internal Server Error: 500)
```

Resolution: The build script uses Docker to collect the Python dependencies that are needed by the plug-in. For Docker to operate correctly, it needs access to the Windows drive that the build script resides in. For information, see [Configure shared volume on Docker for Windows](#).

4. The `mvn` package `-PcollectDependencies` build fails with the following message:

```
[INFO] --- docker-maven-plugin:0.31.0:start (start-container) @ sample-ipam --- [ERROR] DOCKER> Error occurred during container startup, shutting down... [ERROR] DOCKER> I/O Error [Unable to create container for [ipam-dependency-collector:latest] : {"message":"Conflict. The container name "/ipam-dependency-collector-1" is already in use by container "2bfb215381514cd6496ecd5d0103da0a4d94034c5691b25bdf27b16bd2236022". You have to remove (or rename) that container to be able to reuse that name."} (Conflict: 409)]
```

Resolution: Run the following command:

```
run docker ps -a
```

The output should be similar to the following:

```
| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES | -
----- | ----- | ----- | ----- | ----- | ----- | ----- | |
2bfb21538151 | d886e9bba96e | "/bin/sh -c 'yes | c..." | 3 minutes ago | Exited (0) 3 minutes ago | | ipam-dependency-collector-1
```

Locate the container named `ipam-dependency-collector-*` and remove it by using the following command:

```
run docker rm -f 2bfb21538151
```

5. The `mvn` package `-PcollectDependencies` build fails with the following message:

```
[INFO] --- docker-maven-plugin:0.31.0:build (build-image) @ sample-ipam --- [INFO] Building tar: ...\\sample-abx-integration\\target\\docker\\ipam-dependency-collector\\latest\\tmp\\docker-build.tar [INFO] DOCKER> [ipam-dependency-collector:latest]: Created docker-build.tar in 214 milliseconds [ERROR] DOCKER> Unable to build image [ipam-dependency-collector:latest] : "The command '/bin/sh -c tdnf install -y python3-3.6.5-1.ph2 python3-pip-3.6.5-1.ph2 shadow && pip3 install --upgrade pip setuptools && pip3 install certifi' returned a non-zero code: 127"
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
[ "The command '/bin/sh -c tdnf install -y python3-3.6.5-1.ph2 python3-pip-3.6.5-1.ph2 shadow && pip3 install --upgrade pip setuptools && pip3 install certifi' returned a non-zero code: 127" ]
```

Resolution: The **tdnf** Photon OS package manager may have failed to install Python3 due to connectivity errors. Retry after a couple of minutes. If the issue persists, check your internet connection.

Considerations and tips

Knowing the input/output format for each of the 8 operations (4 required and 4 optional) is helpful when implementing the action scripts. The skeleton implementation in the sample IPAM .zip is meant to be a reference tool. Use it at will and modify it as needed.

- You should build, upload, and test your IPAM package after implementing each operation. You can deploy the newly created IPAM .zip into Automation Assembler by selecting Infrastructure → Connections → Integrations → Add Integration → IPAM → Manage IPAM providers → Import Provider Package.
- If everything is configured correctly you should be able to see the sample IPAM in the *Providers* list in the user interface. Select the newly added IPAM provider. The custom form defined in **endpoint-schema.json** should be rendered in the user interface. Complete all mandatory fields and select a running environment (Extensibility On-Prem or for a public cloud such as AWS Lambda or Azure Functions) as described in the product documentation.
 - VMware Aria Automation (on-prem) -- [Use case: How do I configure a provider-specific external IPAM integration](#)
 - VMware Aria Automation (SaaS) -- [Use case: How do I configure a provider-specific external IPAM integration](#)
- Click *Validate* to trigger a new ABX action run. In a new tab, click Extensibility → Activity → Action Runs → Integration Runs. You should be able to see the newly created action run for the Validate Endpoint operation. Click on the action run to see the *inputs* it was called with.
- The `context.request()` can be used to execute REST requests against Automation Assembler. This is used for obtaining the auth credentials.
- The **Deallocate IP** operation contract requires that it must finish with success when asked to deallocate a non-existing IP address. This prevents failure in case an attempt is made from Automation Assembler to deallocate the same IP twice.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

- The **Allocate IP** operation can handle multiple IP allocations in a single request. Failure in one of the consecutive allocations results in failure for the whole operation. Because the IPAM service is not transactional, the script is responsible for rolling back already allocated IPs.

You can implement the IPAM SDK so that integration authentication is performed by using a certificate instead of by using a username and password. You configure the SDK to work with certificate-based authentication by altering the `endpoint-schema.json` file as follows:

1. Add `"state": {"visible": false}` in the `privateKeyId` field description to hide it from the custom form.
2. Change the `"privateKey"` field display type from `"passwordField"` to `"textArea"`.
3. Add a `"default"` value under the Schema definition for Username so that it is populated with a default value of a non-null `privateKeyId`.
4. Change the `privateKey` label to `"Certificate"`.
5. Rebuild the IPAM package to update the custom form for certificate-based authentication.
Note: You can ignore the `auth_credentials["privateKeyId"]` field.

When using certificate-based authentication, the certificate is encrypted with the VMware Aria Automation private key and stored in VMware Aria Automation database.

Learn more – Videos and Documentation

For an **IPAM SDK video** that describes how to use the VMware Aria Automation IPAM SDK to build an IPAM plug-in for your IP address management needs, see [Using the VMware Aria Automation IPAM SDK video](#) on the VMware Cloud Management YouTube channel.

For **provider-specific use case documentation** that illustrates how to use IPAM integration in VMware Aria Automation, see the *Provider-specific external IPAM integration use case* in the Using and Managing Automation Assembler publication for [VMware Aria Automation \(SaaS\)](#) or for your [VMware Aria Automation \(on-prem\)](#) release. For a **provider-specific use case video**, see the [Infoblox IPAM Plug-in 1.1 Integration video](#) on the VMwareLab YouTube channel.

Appendix A – Format for endpoint-schema.json

This appendix describes the form definition format for the `endpoint-schema.json` file contents.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

The form definition is a declarative means expressing the user interface controls used to render form fields, the validations to perform, form field values dependencies, where predefined list of values are retrieved from, and so on. The form renderer generates HTML and JavaScript. Scripts can also include calls to VMware Aria Automation backend services for retrieving dynamic data.

Custom Form Definition (Updated)

```
{
  "layout": {
    "pages": [{
      "id": "page_1",
      "title": "First page title",
      "sections": [{
        "id": "seccion_1",
        "fields": [{
          "id": "field_1"
          "display": ...
          ...
        }, {
          "id": "field_2"
        }]
      }, {
        "id": "section_2",
        "fields": []
      }]
    }, {
      "id": "page_2",
      "title": "Second page title",
      "sections": []
    }]
  },
  "schema": {
    "field_1": {},
    "field_2": {}
  },
  "options": {
    "externalValidations": [
      ...
    ]
  }
}
```

Layout (Updated)



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

The layout part of the form definition can have pages and/or sections. Each section can be hidden based on a constant or on a field value. Sections usually have one or more input fields, which can initially be hidden or read-only until something on the form is changed.

```
"layout": {
  "pages": [{
    "id": "general",
    "title": "General",
    "sections": [{
      "id": "section_1",
      "fields": [{
        "id": "description",
        "display": "textField"
      }, {
        "id": "reason",
        "display": "textField"
      }]
    }, {
      "id": "section_2",
      "fields": [{
        "id": "deployments",
        "display": "integerField"
      }, {
        "id": "leaseDate",
        "display": "textField"
      }]
    }]
  }, {
    "id": "serviceInformation",
    "title": "Service Information",
    "sections": []
  }]
}
```

UI Controls (Updated)

The display property specifies which UI control is rendered. If not set, the default value is “text.”

Drop-Down	"dropDown"
Checkbox	"checkbox"
Radio Group	"radio"
Data Grid	"datagrid"
MultiSelect	"multiSelect"



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Datetime Picker	"dateTime"
Value Picker	"valuePicker"
ComboBox	"combobox"
Multi-value Picker	"multiValuePicker"
Dual List	"dualList"
Link	"link"
Slider	Not Defined

Submit (Updated)

```
"layout": {
  "pages": [{
    "id": "page_1",
    "title": "First page title",
    "sections": [{
      "id": "seccion_1",
      "fields": [{
        "id": "form_label"
        "display": "text"
        "submit": false
      }]
    }]
  }]
}
```

Decimal step (updated)

The "step" property in the schema specifies the legal number intervals for the decimal field.

```
"schema": {
  "storage": {
    "type": {
      "dataType": "decimal"
    },
    "label": "Storage (GB)",
    "step": 0.1
  }
}
```

Integer step (updated)

The "step" property in the schema specifies the legal number intervals for the integer field.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
"schema": {
  "memory": {
    "type": {
      "dataType": "integer"
    },
    "label": "Memory (MB)",
    "step": 1024
  }
}
```

Image description (updated)

The "description" property in the schema specifies alternate text for an image.

```
"schema": {
  "clarity": {
    "type": {
      "dataType": "string"
    },
    "default": "https://cto.vmware.com/wp-content/uploads/2016/11/Picture1.png",
    "placeholder": "Clarity logo"
  }
}
```

Drop down option label (updated)

The "placeholder" property in the schema specifies a placeholder for the dropdown field.

```
"schema": {
  "vm-size": {
    "type": {
      "dataType": "string"
    },
    "label": "Deployment size:",
    "valueList": [{
      "label": "Small",
      "value": "small"
    }], {
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

        "label": "Medium",
        "value": "medium"
    }, {
        "label": "Large",
        "value": "large"
    }, {
        "label": "Custom",
        "value": "custom"
    }
  ]],
  "placeholder": "Select deployment size..."
}
}

```

Data grid (updated)

The "size" property in the layout specifies the number of items displayed per page. The default is 10.

The "placeholder" property in the schema specifies placeholder text when there is no data displayed.

```

"layout":
  ...
  {
    "id": "properties",
    "display": "datagrid",
    "size": 20
  }

"schema": {
  "properties": {
    "type": {
      "dataType": "complex",
      "fields": [{
        "label": "Name",
        "id": "column_1",
        "type": {
          "dataType": "string"
        }
      }, {
        "label": "Value",
        "id": "column_2",
        "type": {
          "dataType": "string"
        }
      }
    ]
  },
  "placeholder": "No items exist.",

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

    "default": [{
      "column_1": "first_prop",
      "column_2": "first"
    }, {
      "column_1": "second_prop",
      "column_2": "second"
    }]
  }
}

```

Multi-value data picker (updated)

The "size" layout property specifies the number of items displayed per page. The default is 10. The "placeholder" schema property specifies placeholder text when there is no data displayed. The "shortValueName" schema property specifies which fields display in the multi-value picker list.

```

"layout":
  ...
  {
    "id": "networks",
    "display": "multiValuePicker",
    "size": 20
  }

"schema": {
  "networks": {
    "type": {
      "dataType": "complex",
      "fields": [{
        "label": "Name",
        "id": "name",
        "type": {
          "dataType": "string"
        }
      }], {
        "label": "Zone",
        "id": "zone",
        "type": {
          "dataType": "string"
        }
      }],
    {
      "label": "Network Domain",
      "id": "network-domain",
      "type": {
        "dataType": "string"
      }
    }
  }
}

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

    }, {
      "label": "Support Public IP",
      "id": "support-public-ip",
      "type": {
        "dataType": "boolean"
      }
    }
  ]],
  "shortValueName": ["name", "zone"],
  "placeholder": "No items exist.",
  "default": [{
    "name": "appnet-isolated-dev",
    "zone": "us-east-1a",
    "network-domain": "rainpole-dev",
    "support-public-ip": false
  },
  {
    "name": "public-subnet-us-east-1b",
    "zone": "us-east-1b",
    "network-domain": "Public VPC",
    "support-public-ip": true
  }
  ]
}
}
}

```

Constant Visible State (Updated)

```

{
  "id": "cpu",
  "display": "integerField",
  "state": {
    "visible": false
  }
}

```

Expression Visible State (Updated)

```

[ {
  "id": "environment",
  "display": "dropDown"
},
{
  "id": "backup-options",
  "display": "dropDown",
  "state": {

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

        "visible": [{
            "equals": {
                "environment": "production"
            },
            "value": true
        }]
    }
}]

```

Constant Read Only State (Updated)

```

{
    "id": "vsphere-tag",
    "display": "textField",
    "state": {
        "read-only": true
    }
}

```

Schema

The "schema" part of the form definition describes the input data that is expected to be entered in the form. The schema consists of a list of fields with the following properties:

- label - label for the input field
- description - short text that will be displayed as a tooltip
- signpost - additional information that will be displayed in a separate popup box
- type - data type of the field
- default - default value
- valueList - a list of predefined values
- constraints - field constraints
 - required
 - *min-value*
 - *max-value*
 - *pattern*
 - *match*

Schema (Updated)

```

"schema":{
    "description": {
        "label": "Description",
        "type": {

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

        "dataType": "string"
    },
    "reason": {
        "label": "Reason for request",
        "signpost": "Specify a reason for this request.",
        "type": {
            "dataType": "string"
        },
        "constraints": {
            "required": true
        }
    },
    "deployments": {
        "label": "Deployments",
        "type": {
            "dataType": "integer"
        },
        "default": 1,
        "constraints": {
            "required": true,
            "min-value": 1,
            "max-value": 100
        }
    },
    "leaseDate": {
        "label": "Lease Date",
        "type": {
            "dataType": "string"
        }
    }
}

```

Signpost

The “signpost” property is the help content that appears in a popup box. Signpost can be plain text or HTML that contains text, links, and images.

```
"signpost": "<a href='https://vmware.github.io/clarity/'>Clarity</a>"
```

```
"signpost": "Plain text"
```

Field data type



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2015-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

The field data “type” property value can be string, integer, decimal, boolean, secure string, complex, or reference. If not set, the default value is string. Use the “type” property when calculating expression field state or constraints.

Field data type	Field value
String	<code>"type": { "dataType": "string"}</code>
Integer	<code>"type": { "dataType": "integer"}</code>
Decimal	<code>"type": { "dataType": "decimal"}</code>
Boolean	<code>"type": { "dataType": "boolean"}</code>
Secure string	<code>"type": { "dataType": "secureString"}</code>
Date Time	<code>"type": { "dataType": "dateTime"}</code>
Multiple	<code>"type": { "dataType": "string", "isMultiple": true}</code>
Complex	<code>"type": { "dataType": "complex", "fields": []}</code>
Reference	<code>"type": { "dataType": "reference", "referenceType": "AD:User"}</code>

Operators

Each field state type supports different operator types.

Field type	Operator type
String, Secure string, Reference	"equals", "notEqual", "contains", "endsWith", "startsWith", "within"
Integer, Decimal, Date Time	"equals", "notEqual", ">", ">=", "<", "<="
Boolean, Complex	"equals", "notEqual"
Multiple	"contains"

Field Constraints



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

Each field can have multiple constraints that are evaluated in the rendered form:

- required
- min-value
- max-value
- pattern
- match

Deployments field constraints

```
"fields": {
  "numberOfDeployments": {
    "type": {
      "dataType": "integer"
    },
    "constraints": {
      "required": "true",
      "min-value": "1",
      "max-value": {
        "type": "scriptAction",
        "id": "com.vmware.vra.endpoint.azure/getReservationLimit"
      }
    }
  },
  "email": {
    "type": {
      "dataType": "string"
    },
    "constraints": {
      "required": "true",
      "pattern": {
        "value": "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$",
        "message": "Must be valid e-mail address."
      }
    }
  },
  "confirm-email": {
    "label": "Confirm Email:",
    "constraints": {
      "match": "email"
    }
  }
}
```



Constant Required Constraint

```
"deployments": {
  "label": "Deployments",
  "type": {
    "dataType": "integer"
  },
  "constraints": {
    "required": true
  }
}
```

Expression Required Constraint (Updated)

```
"reason": {
  "label": "Reason for request",
  "constraints": {
    "required": [{
      ">": {
        "deployments": 10
      },
      "value": true
    }]
  }
}
```

Constant min-value Constraint

A min-value constraint can be set with constant or expression value. If field type is string min-value will specify minimum value of characters length.

```
"password": {
  "label": "Password",
  "type": {
    "dataType": "secureString"
  },
  "constraints": {
    "min-value": 8
  }
}
```

Expression min-value Constraint (Updated)

A min-value constraint can be set with constant or expression value. If field type is string min-value will specify minimum value of characters length.



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
"deployments": {
  "label": "Deployments:",
  "type": {
    "dataType": "integer"
  },
  "constraints": {
    "min-value": [{
      "equals": {
        "environment": "production"
      },
      "value": 4
    }]
  }
}
```

Constant max-value Constraint

A max-value constraint can be set with constant or expression value. If field type is string max-value will specify maximum value of characters length.

```
"description": {
  "label": "Description",
  "type": {
    "dataType": "string"
  },
  "constraints": {
    "max-value": 50
  }
}
```

Expression max-value Constraint

A max-value constraint can be set with constant or expression value. If field type is string max-value will specify maximum value of characters length.

```
"deployments": {
  "label": "Deployments:",
  "type": {
    "dataType": "integer"
  },
  "constraints": {
    "max-value": [{
      "equals": {
        "environment": "test"
      },
    },
  ]
}
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

        "value": 2
      }
    ]
  }
}

```

Pattern constraint

A pattern constraint is an object with two properties - value and message. Value property is constant regular expression value. Message property is custom error message which will be displayed in the tooltip in case of error.

```

"email": {
  "label": "Email",
  "constraints": {
    "pattern": {
      "value": "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$",
      "message": "Must be valid e-mail address."
    }
  }
}

```

Match constraint

A match constraint can be set with a constant value. This value must be other field id.

```

"password": {
  "label": "Password"
},
"confirm-password": {
  "label": "Confirm password:",
  "constraints": {
    "match": "password"
  }
}

```

Constant default value

The field "default" value can be constant value, option from valueList property, bind to other field value or string concatenation. Default values can also be the result of add, subtract or multiply operations.

```

"cpu": {
  "label": "CPU"
  "type": {
    "dataType": "integer"
  }
}

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
    },
    "default": "2"
  }
}
```

Default value from valueList (Updated)

```
"environment": {
  "label": "Environment:",
  "valueList": [{
    "label": "Development",
    "value": "development"
  }, {
    "label": "Test",
    "value": "test"
  }, {
    "label": "Production",
    "value": "production"
  }],
  "default": "production" // Set default value from predefined value list
}
```

Bind default value (Updated)

```
"storage": {
  "label": "Storage",
  "type": {
    "dataType": "integer"
  },
  "default": {
    "bind": "cpu" //Bind storage value to cpu field
  }
}
```

String concatenation

Note: When the value is surrounded with back quotes (for example, "`_machine`") the value is constant.

```
"vsphere-tag": {
  "label": "Tag:",
  "default": {
    "bind": {
      "values": ["environment", "`_machine`"] // Bind field value with
string concatenation
    }
  }
}
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```
    }
  }
```

Add operation (Updated)

```
"sum-field": {
  "label": "Sum:",
  "default": {
    "bind": {
      "values": [ "field_1", "field_2", "`100`"],
      "operator": "add"
    }
  }
}
```

Subtract operation (Updated)

```
"subtract-field": {
  "label": "Difference:",
  "default": {
    "bind": {
      "values": [ "field_1", "field_2"],
      "operator": "subtract"
    }
  }
}
```

Multiply operation (Updated)

```
"storage": {
  "type": {
    "dataType": "decimal"
  },
  "label": "Storage (GB)",
  "default": {
    "bind": {
      "values": [ "storage_mb", "`0.001`"],
      "operator": "multiply"
    }
  }
}
```

vRO Action (Updated)

```
"memory": {
```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

    "type": {
      "dataType": "integer"
    },
    "default": {
      "type": "scriptAction",
      "id": "com.vmware.vra.endpoint.azure/getDefaultMemorySize"
    }
  }
}

```

Value lists (updated)

Lists of values can be assigned to a form field as either static (value is specified in the field schema) or dynamic (value is retrieved from an external data source like a VMware Aria Automation database, VMware Aria Automation Orchestrator action or a REST API service). To populate the `valueList` property with an Automation Orchestrator action, the return type must be **array of string**, **properties** or **array of properties**.

Deployment size values – static (Updated)

```

"fields": [
  "deploymentSize": {
    "type": {
      "dataType": "string"
    },
    "valueList": [{
      "label": "Small",
      "value": "small"
    }, {
      "label": "Medium",
      "value": "medium"
    }, {
      "label": "Large",
      "value": "large"
    }, {
      "label": "Custom",
      "value": "custom"
    }
  ]
}
}

```

Deployment size values – dynamic (Updated)

```

"fields": {
  "deploymentSize": {
    "type": {

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

    "dataType": "string"
  },
  "valueList": {
    "type": "scriptAction",
    "id": "com.vmware.vra.endpoint.azure/getVmSizes",
    "parameters": [
      {
        "param_1": "field1",
        "$type": {
          "dataType": "boolean"
        }
      },
      {
        "param_2": "`2`",
        "$type": {
          "dataType": "integer"
        }
      },
      {
        "param_3": "field2",
        "$type": {
          "dataType": "string"
        }
      }
    ]
  }
}

```

Options

The "options" part of the form additional form functionalities and it is not required unless these functionalities are used. The options consist of the following properties:

- externalValidations - list of external validations applied on form
 - label
 - source
 - type - type of the external source
 - id- unique identifier of the external source
 - parameters - parameters for the external source
 - target - array of fields in which the returned error message is applied (optional)

```

"options": {
  "externalValidations": [
    {

```



Creating and deploying a provider-specific IPAM integration package for VMware Automation Assembler

```

    "label": "Amazon form validation",
    "source": {
      "type": "scriptAction",
      "id": "com.zzz.vra.endpoint.aws/myFormExternalValidation1",
      "parameters": []
    },
    "target": []
  },
  {
    "label": "Amazon field validation with params",
    "source": {
      "type": "scriptAction",
      "id":
"com.zzz.vra.endpoint.aws/myInputExternalValidationWithParams",
      "parameters": [
        {
          "param_1": "field1",
          "$type": {
            "dataType": "boolean"
          }
        },
        {
          "param_2": "`2`",
          "$type": {
            "dataType": "integer"
          }
        },
        {
          "param_3": "field2",
          "$type": {
            "dataType": "string"
          }
        }
      ]
    },
    "target": [
      "field_1",
      "field_1"
    ]
  }
]
}

```

