

Expanding the Reach: Leverage Infoblox APIs to Drive Automation Across Your Enterprise

John Voss

Principal Product Manager, Solutions



Agenda

- Overview of Infoblox APIs/System Interfaces
- Overview of Infoblox REST API (WAPI)
- WAPI key concepts
- WAPI examples
- Advanced topics/tips and tricks
- Q&A



Infoblox Actionable Network Intelligence Platform

- Centralized IP address information, lease history, DNS query information, network configuration properties, DNS security events, and other core network services data improves
 - IT operations and asset management
 - Network and system performance
 - Time to service through automation
 - System troubleshooting and outage response
 - Security and compliance
 - Incidence response and mitigation
 - Visibility
- Metadata (extensible attributes) increases business agility
 - Linking network data to business groups and/or business functions/applications simplifies management and maintenance and creates operational consistency

Infoblox APIs are the Keys to Unlocking this Business Value



Infoblox System Interfaces

Type	Description
PERL and RESTful API	Full access to Infoblox distributed database objects via PERL and RESTful APIs
Syslog	Rich set of DDI data available including DNS query data, RPZ hits, IP Address allocations/de-allocations, and Infoblox audit data (object adds, changes, deletes)
CISCO ISE (pxGrid)	Inbound and outbound pxGrid messaging to support <ul style="list-style-type: none">• Enforcement of NAC policy using CISCO ISE based on indicator of compromise published by Infoblox• Data Enrichment use-cases for IB and CISCO ISE
STIX/TAXII	Inbound STIX/TAXII support for receiving Indicators of Compromise (IOC) used for blocking bad domains via DNS Firewall rules (RPZ)
SNMP	SNMP support for events and variables including Grid system information and DNS and DHCP statistics. Examples of traps are notification on DNS security events and DHCP Lease or Fixed Address conflicts
E-Mail	Support for E-mail notifications for events and system status including Grid system information and IPAM, DNS and DHCP events
Discovery/vDiscovery	Ability to find and identify devices on the network and pull IPAM and metadata from VMware, AWS, and OpenStack environments



Infoblox Data Available via APIs

Data	Uses
IP Address Information	Asset Management, Configuration Management, Network Monitoring/Management, Compliance, Security NAC/Device Scanning
User Identity	Asset Management, Configuration Management, Network Monitoring/Management, Compliance, Security NAC/Device Scanning
DHCP Lease	Device Information (DHCP Fingerprinting), Auditing/Compliance, Security NAC/Device Scanning
Network/Device Discovery Data	Asset Management, Configuration Management, Network Monitoring/Management, Compliance
DNS Query	Auditing/Compliance, Performance Management, Network Optimization
DNS Security	Security Alerts and Mitigation.
Extensible Attributes (Metadata)	User defined data mapping network information to business usage. Asset Management, Configuration Management, Network Monitoring/Management, Compliance, Security NAC/Device Scanning



Infoblox REST API (WAPI)

- Easy to use REST API for creating, reading, updating, or deleting data and system configurations in Infoblox Grid
- Ability tag any object with user defined metadata (Extensible Attributes)
 - E.g., network 10.10.10.0/24 is in New York Data Center, network 20.20.20.0/24 is in New Jersey Sales Office
- Ability to search for data by Extensible Attributes
 - E.g., Find all networks allocated to New York Data Center
- Roles based access control with ability to restrict access to specific objects and specific data



Infoblox REST API (WAPI) cont.

- WAPI = Web API
- Simple to use Representational State Transfer (REST) API providing a Create, Read, Update, Delete interface to Infoblox database objects and system configurations
- Basic object interactions use HTTP methods (GET, POST, PUT, DELETE)
- All API calls are encrypted using SSL/TLS and authenticated using HTTP basic authentication with NIOS Admin user credentials
 - Support for sessions using cookies
- Support for some functions (e.g. get next available IP and get next available network)
- Some synthetic objects available for querying/reading multiple object types (e.g. Allrecords)
- Ability to send multiple API requests in a single HTTPS POST using the Object Body Request API



WAPI Key Concepts

- WAPI APIs are based on Infoblox objects
 - E.g. to update host records, you call record:host. To update A records, you call record:a, PTR records, record:ptr, etc.
- To update or delete a particular record/object, you must know the Reference of the object
 - This is done by making a separate GET/query API call to get the object reference
 - After getting the object reference, updates can be done via PUT and deletes via DELETE
- WAPI APIs can take various parameters like max results and returned objects
 - Max results controls how many records can be returned in a single API call
 - By default max results = -1000 which means that if there are more than 1000 records in the results an error is returned
 - A positive number for max results (e.g., 1000) means only the first 1000 records will be returned
 - Return fields controls which fields/properties are returned in the API call response
 - Not all fields are returned by default. Additional fields need to be requested using `_return_fields%2b` (`_return_fields+` escaped for HTTP URI transmission)



WAPI Key Concepts

- Permissions
 - To use WAPI, the NIOS Admin User making the API call must have the correct object permissions
 - If a user does not have permissions to a given object, then GET requests will return no results rather than an error
- Searches
 - To search for Extensible Attributes (EAs), put * before the EA name
 - E.g. to search on EA "State" use *State=CA
 - All searches are case sensitive by default. To make a search case insensitive, add:
 - E.g. *State:=California will match both "California" and "california" but *State=CA will only match "California" and not "california"
 - Regex searches are supported using ~=
 - Searches for multiple properties are always "and" searches. I.e., *VLAN=100,*State=CA will match records that have both these EA names/values set, but will not match if just one property is set



WAPI Key Concepts

- Updating Extensible Attributes (EAs)
 - To add or update a specific EA use extattrs+ in your PUT. Otherwise the extattrs you specify will replace any other extattrs not included in your PUT.
 - E.g. PUT `"extattrs":{"State":{"value":"CA"}}` would replace all other EAs on the object
 - PUT `"extattrs+":{"State":{"value":"CA"}}` will only add the State EA if it doesn't already exist or update it to the new value if it does exist without replacing the other EAs already on the record
- Host Records
 - Can perform many different roles
 - IPAM Only, DNS Only, IPAM + DHCP, IPAM + DNS, IPAM + DNS + DHCP
 - Support multiple IP addresses on a single Host record
 - All IP addresses on the Host record share the same A record if DNS is enabled
 - Individual IP addresses can be added/removed from a Host record using PUT with `ipv4addr+` or `ipv4addr-`
 - A/PTR records associated with a Host record do not show up when querying `record:a` or `record:ptr` objects
 - `allrecords` can be used instead to get all the dns records for a given zone
 - Aliases do not show up when querying the `record:cname` object
 - Even though they serve the same function, aliases on a host record must get GET/set via the `record:host` object and not via the `record:cname` objects
 - `allrecords` can be used instead to get both aliases and cnames for a particular zone



WAPI Examples

- Creating Network
- Creating Host Record with Get Next Available IP Function
- Adding EA names/values when Creating Records
- Updating Host Record
- Changing EA values on Host Records
- Deleting Host Records



WAPI Examples

- Create Network 1.2.3.0/24 in the default Network View

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v1.4/network -d  
'{"network":"1.2.3.0/24","network_view":"default"}'  
"network/ZG5zM5ldHdvcmskMS4yLjMuMC8yNC8w:1.2.3.0/24/default"
```

- Create Network 2.3.4.0/24 with EA Site = Virginia

```
curl -k1 -u ibuser:ibuser -X POST https://10.10.100.81/wapi/v1.4/network -H "Content-type:application/json" -d  
'{"network":"2.3.4.0/24","extattrs":{"Site":{"value":"Virginia"}}}'  
"network/ZG5zM5ldHdvcmskMi4zLjQuMC8yNC8w:2.3.4.0/24/default"
```



WAPI Examples

- Get Network with EA Site = Virginia

```
curl -k1 -u ibuser:ibuser -X GET https://10.10.100.81/wapi/v1.4/network?*Site=Virginia
```

```
[  
  {  
    "_ref": "network/ZG5zLm5ldHdvcmskMi4zLjQuMC8yNC8w:2.3.4.0/24/default",  
    "network": "2.3.4.0/24",  
    "network_view": "default"  
  }  
]
```



WAPI Examples

- Create Host Record with Get Next Available IP

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v1.4/record:host?_return_fields=ipv4addrs -d '{"name":"host1.somedomain.com","ipv4addrs":[{"ipv4addr":"func:nextavailableip:10.1.1.0/24"}]}'
```

```
{
  "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qx:host1.somedomain.com/default",
  "ipv4addrs": [
    {
      "_ref":
"record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4uaG9zdDEuMTAuMS4xLjQ3Lg:10.1.1.47/host1.somedomain.com/default",
      "configure_for_dhcp": false,
      "host": "host1.somedomain.com",
      "ipv4addr": "10.1.1.47"
    }
  ]
}
```



WAPI Examples

- Create Host Record with Get Next Available IP and Setting EA Location

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v1.4/record:host?_return_fields=ipv4addrs,extattrs -d '{"name":"host2.somedomain.com","ipv4addrs":[{"ipv4addr":{"func":nextavailableip:10.1.1.0/24}}],"extattrs":{"Location":{"value":"Building 1"}}}'
```

```
{
  "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
  "extattrs": {
    "Location": {
      "value": "Building 1"
    }
  },
  "ipv4addrs": [
    {
      "_ref":
"record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4uaG9zdDluMTAuMS4xLjQ4Lg:10.1.1.48/host2.somedomain.co
m/default",
      "configure_for_dhcp": false,
      "host": "host2.somedomain.com",
      "ipv4addr": "10.1.1.48"
    }
  ]
}
```



WAPI Examples

- Update Host Record with alias (i.e., CNAME)

curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X GET <https://10.10.100.81/wapi/v1.4/record:host?name=host2.somedomain.com>

```
[
  {
    "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
    "ipv4addrs": [
      {
        "_ref":
"record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4uaG9zdDIuMTAuMS4xLjQ4Lg:10.1.1.48/host2.somedomain.co
m/default",
        "configure_for_dhcp": false,
        "host": "host2.somedomain.com",
        "ipv4addr": "10.1.1.48"
      }
    ],
    "name": "host2.somedomain.com",
    "view": "default"
  }
]
```

curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X PUT https://10.10.100.81/wapi/v1.4/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy?return_fields=aliases -d

```
'{"aliases":["www.someotherdomain.com"]}'
{
  "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
  "aliases": [
    "www.someotherdomain.com"
  ]
}
```



WAPI Examples

- Change EA Value on Host Record

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X GET 'https://10.10.100.81/wapi/v1.4/record:host?name=host2.somedomain.com&_return_fields%2b=extattrs'
```

```
[
  {
    "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
    "extattrs": {
      "Location": {
        "value": "Building 1"
      }
    },
    "ipv4addrs": [
      {
        "_ref":
"record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4uaG9zdDIuMTAuMS4xLjQ4Lg:10.1.1.48/host2.somedomain.com/default",
        "configure_for_dhcp": false,
        "host": "host2.somedomain.com",
        "ipv4addr": "10.1.1.48"
      }
    ],
    "name": "host2.somedomain.com",
    "view": "default"
  }
]
```

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X PUT
https://10.10.100.81/wapi/v2.1/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy?_return_fields=extattrs -d '{"extattrs":{"Location":{"value":"Building 2"}}}'
```

```
{
  "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
  "extattrs": {
    "Location": {
      "value": "Building 2"
    }
  }
}
```



WAPI Examples

- Delete Host Record

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X GET 'https://10.10.100.81/wapi/v1.4/record:host?name=host2.somedomain.com'
```

```
[
  {
    "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default",
    "ipv4addrs": [
      {
        "_ref":
"record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4uaG9zdDluMTAuMS4xLjQ4Lg:10.1.1.48/host2.somedomain.co
m/default",
        "configure_for_dhcp": false,
        "host": "host2.somedomain.com",
        "ipv4addr": "10.1.1.48"
      }
    ],
    "name": "host2.somedomain.com",
    "view": "default"
  }
]
```

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X DELETE
```

```
https://10.10.100.81/wapi/v2.1/record:host/ZG5zLmhvc3QkLI9kZWZhdx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:record:host/ZG5zLmhvc3QkLI9kZWZhdx0LmNvbS5zb21lZG9tYWluLmhvc3Qy:host2.somedomain.com/default
```



Advanced Topics/Tips and Tricks

- Creating Host Records with DNS and/or DHCP Disabled
- Microsoft Synch
 - Using Fixed Addresses, A/PTR Records instead of Host records
- Making multiple API Calls in a single POST using Object Body API



Advanced Topics/Tips and Tricks

- Creating Host Records with DNS and/or DHCP Disabled
 - Host records can be used in the following ways:
 - IPAM Only - dhcp_enabled = false, dns_enabled = false
 - IPAM + DNS - dhcp_enabled = false, dns_enabled = true (Default)
 - IPAM + DHCP - dhcp_enabled = true, dns_enabled = false
 - IPAM + DNS and DHCP - dhcp_enabled = true, dns_enabled = true
 - To control this setting, set the corresponding attribute when creating the Host record. By default, dns_enabled is set to true and dhcp_enabled is false



Advanced Topics/Tips and Tricks

- Creating Host Records with DNS and/or DHCP Disabled

- Default Example (DNS Enabled, DHCP Disabled)

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST
```

```
https://10.10.100.81/wapi/v1.4/record:host?return\_fields%2b=ipv4addrs,configure\_for\_dns -d
```

```
{ "name": "host3.somedomain.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip:10.1.1.0/24"}] },
```

```
{
```

```
  "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5zb21lZG9tYWluLmhvc3Qz:host3.somedomain.com/default",
```

```
  "configure_for_dns": true,
```

```
  "ipv4addrs": [
```

```
    {
```

```
      "_ref":
```

```
"record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnNvbWVkb21haW4uaG9zdDMuMTAuMS4xLjQ4Lg:10.1.1.48/host3.somedomain.com/default",
```

```
      "configure_for_dhcp": false,
```

```
      "host": "host3.somedomain.com",
```

```
      "ipv4addr": "10.1.1.48"
```

```
    }
```

```
  ],
```

```
  "name": "host3.somedomain.com",
```

```
  "view": "default"
```

```
}
```



Advanced Topics/Tips and Tricks

- Creating Host Records with DNS and/or DHCP Disabled

- DNS Disabled, DHCP Enabled

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST
https://10.10.100.81/wapi/v1.4/record:host?\_return\_fields%2b=ipv4addrs,configure\_for\_dns -d
'{"name": "host4.somedomain.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip:10.1.1.0/24", "configure_for_dhcp": true, "mac": "00:5
0:56:c0:00:09"}], "configure_for_dns": false}'
{
  "_ref":
  "record:host/ZG5zMmhvc3QkLm5vbl9ETINfaG9zdF9yb290LjAuMTQ2MDcwNjMzMdUxMy5jb20uc29tZWVpbi5ob3N0NA:host4.so
medomain.com/%20",
  "configure_for_dns": false,
  "ipv4addrs": [
    {
      "_ref":
      "record:host_ipv4addr/ZG5zMmhvc3RfYWVWRkcmVzcyQubm9uX0ROU19ob3N0X3Jvb3QuMC4xNDYwNzA2MzMwNTEzLmNvbS5zb21lZ
G9tYWluLmhhvc3Q0LjEwLjEuMS41MC4:10.1.1.50/host4.somedomain.com/%20",
      "configure_for_dhcp": true,
      "host": "host4.somedomain.com",
      "ipv4addr": "10.1.1.50",
      "mac": "00:50:56:c0:00:09"
    }
  ],
  "name": "host4.somedomain.com",
  "view": " "
}
```



Advanced Topics/Tips and Tricks

- Microsoft Synch

- Individual A/PTR and Fixed Address/IPv4 Reservation records must be used with Microsoft Synch instead of Host Records
- IPv4 Reservation records are the same as Fixed Address records but using the mac address "00:00:00:00:00:00"
- Example – create IPv4 Reservation with Next Available IP

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST
https://10.10.100.81/wapi/v1.4/fixedaddress?_return_fields=ipv4addr -d
'{"ipv4addr":"func:nextavailableip:10.1.1.0/24","mac":"00:00:00:00:00:00"}'
{
  "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZHJlc3MkMTAuMS4xLjUyLjAuLg:10.1.1.52/default",
  "ipv4addr": "10.1.1.52"
}
```



10.1.1.52

Used

IPv4 Reservation




Advanced Topics/Tips and Tricks

- Microsoft Synch
 - Example – create A Record

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v1.4/record:a -d '{"name":"host10.somedomain.com","ipv4addr":"10.1.1.52"}' "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQuY29tLnNvbWVkb21haW4saG9zdDEwLDEwLjEuMS41Mg:host10.somedomain.com/default"
```

		10.1.1.52	host10.somedomain.com	Used	IPv4 Reservation, A Record
---	---	---------------------------	-----------------------	------	----------------------------

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v1.4/record:ptr -d '{"ptrdname":"host10.somedomain.com","ipv4addr":"10.1.1.52"}' "record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFkZHIuMTAuMS4xLjUyLmhvc3QxMC5zb21lZG9tYWluLmNvbQ:52.1.1.10.in-addr.arpa/default"
```

		10.1.1.52	host10.somedomain.com	Used	IPv4 Reservation, A Record, PTR Record
---	---	---------------------------	-----------------------	------	--



Advanced Topics/Tips and Tricks

- Object Body Request
 - Added in WAPI 2.1 (NIOS 7.1)
 - Allows you to make multiple API calls in a single HTTPS POST



Advanced Topics/Tips and Tricks

- Object Body Request Example – Request

```
curl -k1 -u ibuser:ibuser -H "Content-Type: application/json" -X POST https://10.10.100.81/wapi/v2.1/request -d '
```

```
[{
  "method": "POST",
  "object": "fixedaddress",
  "data": {
    "ipv4addr": "func:nextavailableip:10.1.1.0/24","mac":"00:00:00:00:00:00"
  },
  "assign_state": {
    "ipaddress": "ipv4addr"
  },
  "args": {
    "_return_fields": "ipv4addr"
  },
  "enable_substitution": true,
  "discard": false
},
{
  "method": "POST",
  "object": "record:a",
  "enable_substitution": true,
  "data": {
    "name": "host11.somedomain.com","ipv4addr":"##STATE:ipaddress:##"
  },
  "discard": false
},
{
  "method": "POST",
  "object": "record:ptr",
  "enable_substitution": true,
  "data": {
    "ptrdname": "host11.somedomain.com","ipv4addr":"##STATE:ipaddress:##"
  },
  "discard": false
}
]
```



Advanced Topics/Tips and Tricks

- Object Body Request Example – Result

```
[
  {
    "result": {
      "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZHJlc3MkMTAuMS4xLjU2LjAuLg:10.1.1.56/default",
      "ipv4addr": "10.1.1.56"
    }
  },
  "record:a/ZG5zLmJpbmRfYSQuX2RIZmF1bHQuY29tLnNvbWVkb21haW4saG9zdDEwLjEuMS41Ng:host11.somedomain.com/default",
  "record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHIuMTAuMS4xLjU2Lmhvc3QxMS5zb21lZG9tYWluLmNvbQ:56.1.1.10.in-addr.arpa/default",
]
```

 	10.1.1.56	host11.somedomain.com	Used	IPv4 Reservation, A Record, PTR Record
---	---------------------------	-----------------------	------	--



Q&A



Additional Information/Resources

- Infoblox Experts Community API Forum – https://community.infoblox.com/t5/API-Integration/bd-p/API_Integration
- Infoblox REST Examples – <https://community.infoblox.com/t5/API-Integration/The-definitive-list-of-REST-examples/m-p/1214/highlight/true#M2>
- API/System Integration Technical Advisory Board – <https://community.infoblox.com/t5/API-System-Integrators/gp-p/APISystemIntegrators>
- Searchable On-line WAPI Documentation – [https://\[your grid master ip/fqdn\]/wapidoc](https://[your grid master ip/fqdn]/wapidoc)

