CA PM Multi Tenancy
Workflows and REST calls for automated procedures

Overview

The multi-tenancy feature in CA PM lets you monitor discrete customer environments separately and securely. A tenant represents a customer environment that a managed service provider (MSP) administers. Each tenant environment is independent and effectively functions as a separate instance of CA Performance Management. Each instance can contain multiple users and roles that are not shared among tenants. Each tenant is associated with one or more IP domains. All items in the associated IP domains are associated with that tenant. The global administrator or the tenant administrator sets up the monitoring environment for the tenant.

The following table shows configuration item scope

Global	**Tenant specific**
	IP domain
	SNMP profiles
	Users
	Roles
	Monitoring groups and collections
	Custom dashboards
	Custom menus
	On Demand Reports
	Context page customization
	Discovery profiles
	Threshold profiles
Metric Families, Vendor Certs	
Monitoring Profiles	
Shared Provider Groups	

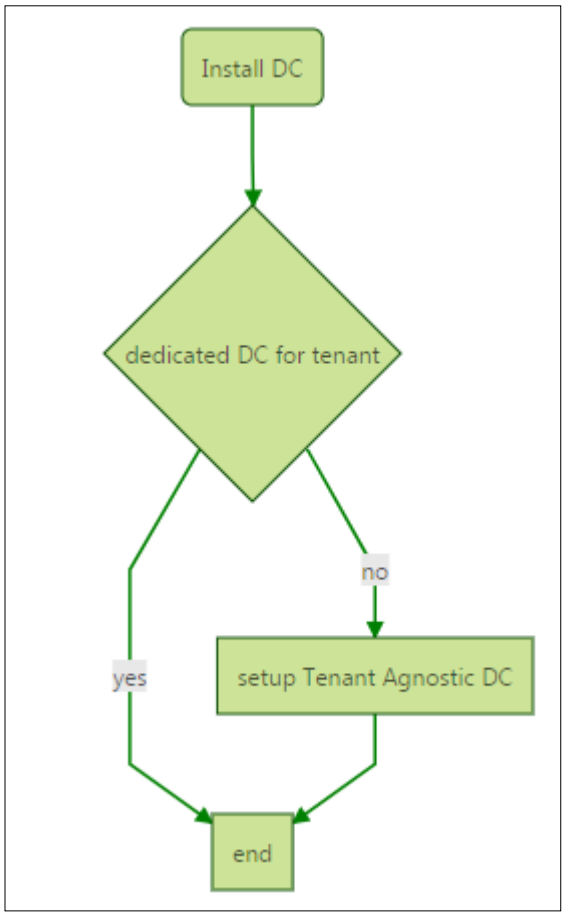
This document describes typical workflows with REST webservice call templates that can be used to setup tenants in an automated procedure.

Basic workflows

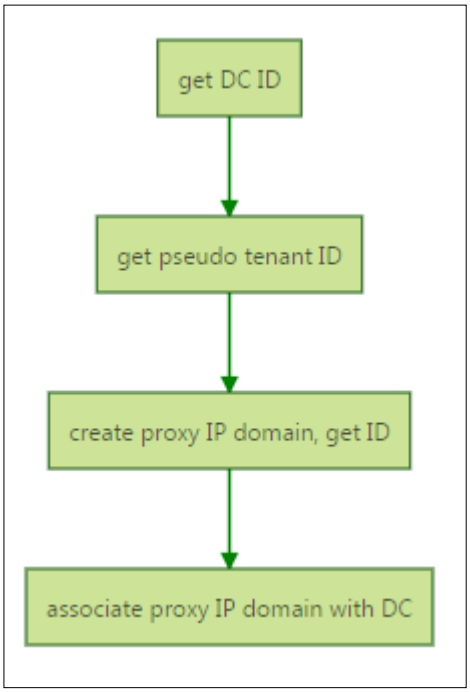
The workflows include

- Preparations
- Optional: Tenant Agnostic DC setup
- Basic Tenant Setup
- Tenant Configuration
- Create Roles and Users
- Create Threshold Profiles
- Share customizations

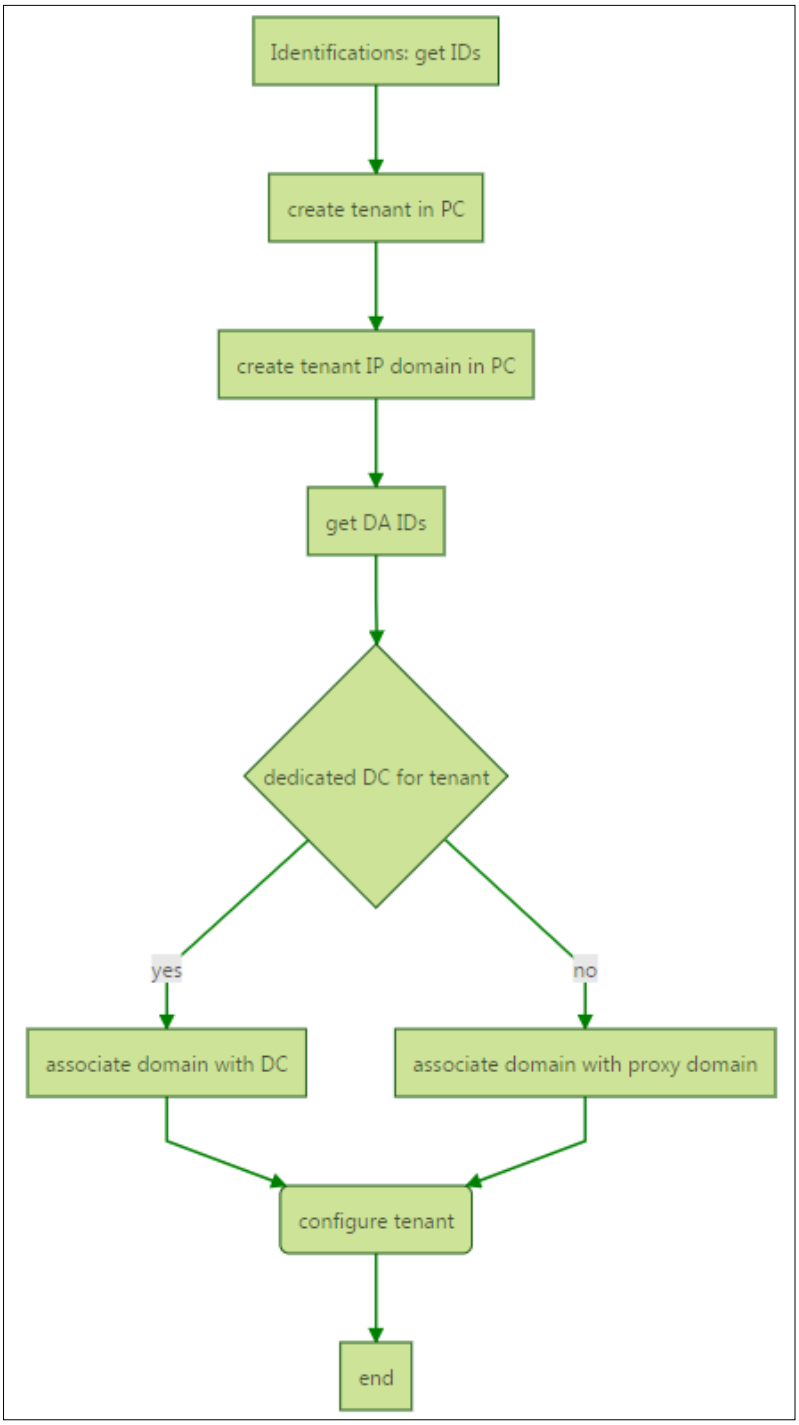
Preparations



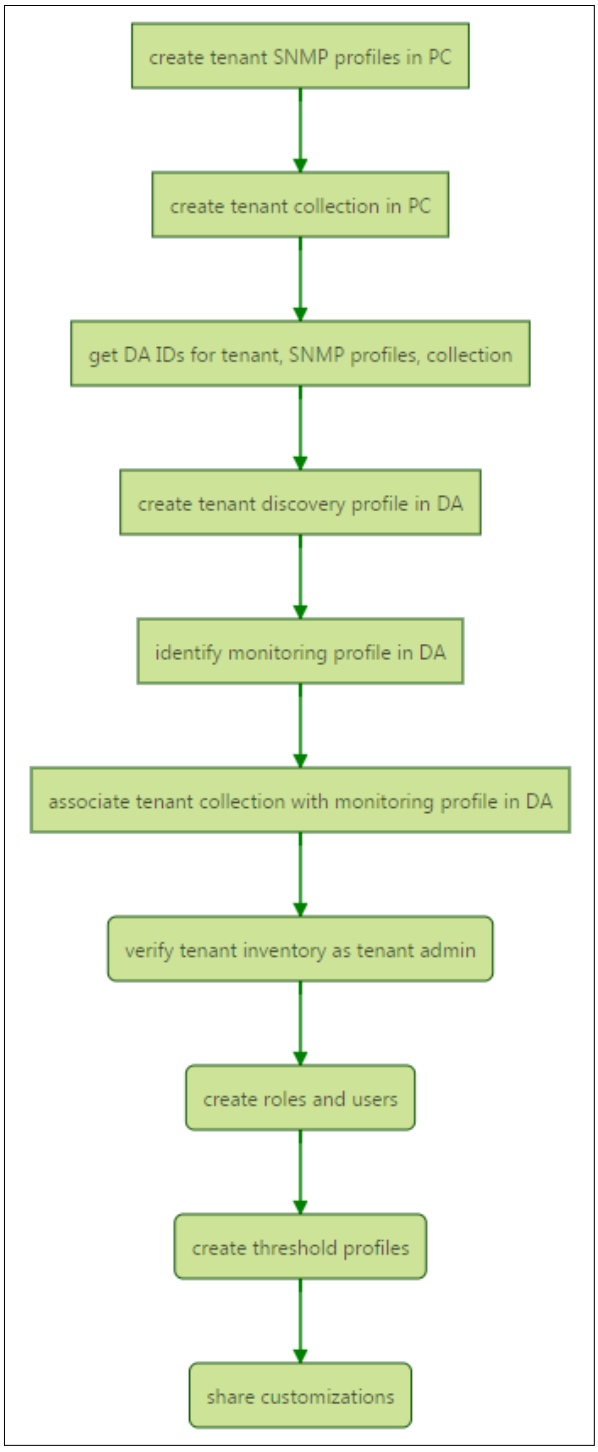
Tenant Agnostic DC



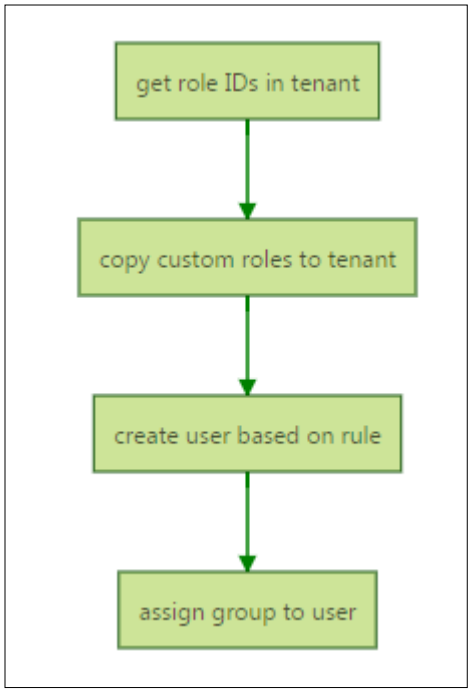
Tenant Basic Setup



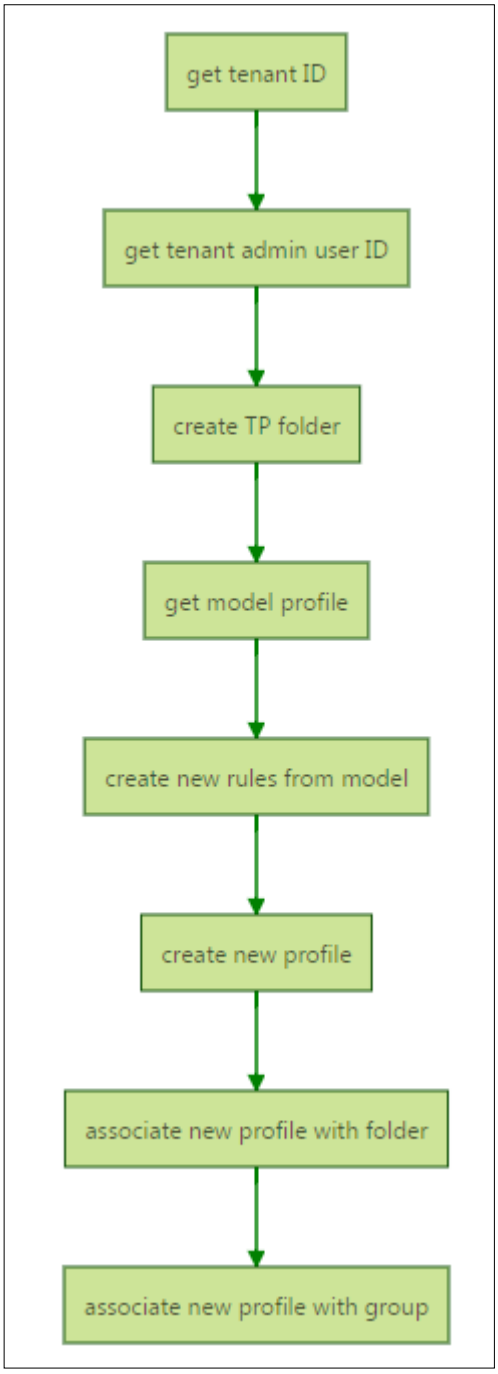
Tenant Configuration



Create Roles and Users



Create Threshold Profiles



Install Data Collector

A Data Collector (DC) can be shared across several tenants (tenant agnostic DC) or dedicated to a single tenant. In any case the DC can be installed with a generic procedure and initial default values. The instantiation as tenant agnostic or specific DC is controlled by DA REST calls as described below.
Reference: https://docops.ca.com/ca-performance-management/3-2/en/installing/install-the-data-collectors

Identifications

Identify basic settings needed for an automated workflow

Topic	Description	Input/output
Host connectivity	Verify connectivity to CA PC and DA REST webservices GET http://pc_host:8181/pc/center/rest Basic Auth User: *global admin* Password: (global admin password) GET http://da_host:8581/rest	
Get DA datasource ID	GET http://pc_host:8181/pc/center/webservice/datasources/ Basic Auth User: *global admin* Password: (global admin password) Get the id value for type DATA_AGGREGATOR -> DS_ID_DA Typically it is 3.	Output: DS_ID_DA
Identify DC	POST http://da_host:8581/rest/dcms/filtered body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <Item.Name type="STARTS_WITH">***{DC_HOSTNAME}***</Item.Name> </And></Filter> <Select use="exclude"> </Select> </FilterSelect> Response: ID → DC_ID	Input: DC_HOSTNAME Output: DC_ID

Setup tenant agnostic Data Collector

This step is optional and applies if the tenant does not require dedicated Data Collector. It must be applied once for each DC that should host several IP domains and tenants.

The procedure is defined in product documentation: https://docops.ca.com/ca-performance-management/3-2/en/administrating/multi-tenancy/tenant-agnostic-data-collectors

Topic	Description	Input/output
Input parameters	Input parameters for this section: DC_ID	
Get Pseudo Tenant ID	GET http://da_host:8581/rest/pseudotenants get the <ID> value → PT_ID	Output: PT_ID
Create proxy IP Domain	POST http://da_host:8581/rest/tenant/PT_ID/proxyipdomains/	Output: PROXY_ID

	Body: <ProxyIPDomain version="0.0.0"> <IPDomain version="0.0.0"> <Name>*Proxy IP Domain*</Name> <Description>*Description for Proxy IP Domain*</Description> <DNSProxyEnabled>false</DNSProxyEnabled> <DNS1Address>0.0.0.0</DNS1Address> <DNS2Address>0.0.0.0</DNS2Address> <DNS1Port>0</DNS1Port> <DNS2Port>0</DNS2Port> </IPDomain> </ProxyIPDomain> Response: ID → PROXY_ID	
Associate proxy IP domain with DC	PUT http://da_host:8581/genericWS/dcm/{DC_ID} Body: <DCM> <IP_DOMAIN_ID>***{PROXY_ID}***</IP_DOMAIN_ID> </DCM>	Input: DC_ID, PROXY_ID

Tenant Setup

Perform this step for each tenant.

The general procedure is defined in product documentation: https://docops.ca.com/ca-performance-management/3-2/en/administrating/multi-tenancy/configure-a-tenant-environment

Topic	Description	Input/Output
This section describes the following aspects: 1. Create tenant 2. Create tenant IP domain 3. Get tenant and IP domain DA IDs 4. Associate domain with DC / proxy domain 5. Create SNMP profiles 6. Get basic group IDs 7. Create collection and group 8. Get SNMP profile DA IDs 9. Create discovery profile 10. Get collection and monitoring profile DA IDs 11. Assign collection to monitoring profile 12. verifications		
Input parameters	Input parameters for this section: TenantName TenantDescription AccountNumber IP Domain Name IP Domain Description DS_ID_DA SNMP Profile Name and credentials (v1/v2c and v3) Discovery Profile Name and details (IP address or host name list) CollectionName Monitoring Profile Name	

Create tenant	POST http://pc_host:8181/pc/center/webservice/tenants/ Basic Auth User: *global admin* Password: (global admin password) Body: <tenant> <tenantName>***TenantName***</tenantName> <tenantDesc>***TenantDescription***</tenantDesc> <accountIdentifier>***AccountNumber***</accountIdentifier> <status>Enabled</status> <removable>false</removable> <theme>CA-Blue</theme> <defaultCulture>en-US</defaultCulture> </tenant> Response: <ID> → TN_ID	Input: TenantName/description, number Output: TN_ID
Create tenant IP domain	POST http://pc_host:8181/pc/center/webservice/domains/ Basic Auth User: *TenantName_admin* Password: (tenant admin password) Body: <domain> <itemDesc>***IP domain description***</itemDesc> <itemName>***IP domain name***</itemName> <TenantID>***{TN_ID}***</TenantID> <primaryDNSAddress>0.0.0.0</primaryDNSAddress> <primaryDNSPort>0.0.0.0</primaryDNSPort> <secondaryDNSAddress>0.0.0.0</secondaryDNSAddress> <secondaryDNSPort>0.0.0.0</secondaryDNSPort> <isDnsProxyEnabled>false</isDnsProxyEnabled> </domain> Response: <ID> in CA PC	Input: TN_ID, IP domain name & description
Re-Sync DA with PC	(before the corresponding DA IDs can be retrieved, a sync is required, await automatic sync or force it explicitly) PUT http://pc_host:8181/pc/center/webservice/datasources/dataSourceId/**DS_ID_DA**/false Basic Auth User: *global admin* Password: (global admin password) Body: <emptybody/>	Input: DS_ID_DA
Get tenant DA ID	POST http://da_host:8581/rest/tenants/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <Tenant.Name type="EQUAL">***TenantName***</Tenant.Name> </And></Filter> <Select use="exclude"></Select> </FilterSelect> Response: <ID> → TN_ID_DA	Input: TenantName Output: TN_ID_DA
Get tenant IP domain DA ID	POST http://da_host:8581/rest/ipdomains/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <IPDomain.Name type="EQUAL">***IP domain name***</IPDomain.Name> </And></Filter> <Select use="exclude"></Select> </FilterSelect>	Input: IP Domain name Output: TNIP_ID_DA

	Response: <ID> → TNIP_ID_DA	
Tenant agnostic DC? Associate with proxy IP domain	PUT http://da_host:8581/rest/ipdomains/**TNIP_ID** Body: <IPDomain version="0.0.0"> <ProxyIPDomain>***PROXY_ID***</ProxyIPDomain> </IPDomain>	Input: TNIP_ID, PROXY_ID
	Verify tenant agnostic DC setup: On DA host: /opt/IMDataAggregator/scripts/dadiag.py audit tadco /opt/IMDataAggregator/scripts/dadiag.py show tadco (will verify the setup and list tenant, domain, DC)	
Dedicated DC for tenant? Associate with DC	PUT http://da_host:8581/genericWS/dcm/**DC_ID** Body: <DCM> <IP_DOMAIN_ID>***TNIP_ID***</IP_DOMAIN_ID> </DCM>	Input: DC_ID, TNIP_ID
	Verify DC setup: GET http://da_host:8581/rest/dcms/**DC_ID** (verify IPDomainID, TenantName, CollectorState)	
Create SNMP profiles	(example: v1/v2c profile) POST http://pc_host:8181/pc/center/webservice/profiles/saveProfile/rankTiesAscendingByDate Basic Auth User: *TenantName_admin* Password: (tenant admin password) Body: <SnmpProfile> <name>**TN_ID**_*name*</name> <port>161</port> <communityString>*community*</communityString> <context/> <version>Version1or2C</version> <securityLevel>NoAuthNoPriv</securityLevel> <authProtocol>None</authProtocol> <authPassword/> <privProtocol>None</privProtocol> <privPassword/> <enabled>true</enabled> <useForSnmpSet>false</useForSnmpSet> <tenantID>***TN_ID***</tenantID> </SnmpProfile>	Input: TN_ID, name, community (name should be unique, e.g. prefix with tenant ID)
Get inventory and group ID	GET http://pc_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FDefined%20Tenants%2F**TenantName** Basic Auth User: *global admin* Password: (global admin password) From response, get id value for name="Inventory" → INV_ID	Input: TenantName Output: INV_ID
Create collection	(example: rule with IP address range) POST http://pc_host:8181/pc/center/webservice/groups/false/false Basic Auth User: ***TenantName***_*admin* Password: (tenant admin password) Body: <GroupTree path="/All Groups/Defined Tenants/***TenantName***/Collections">	Input: TenantName, CollectionName, INV_ID,

	<Group name="***CollectionName***" desc="***TenantName CollectionName***" inherit="true" type="user group"/> <Rules allowDeletes="true" saveRules="true"> <Rule add="device" itemSubTypeName="" itemTypeLabel="Device" itemTypeLabels="Devices" name="Add Devices"> <Match> <Compare readOnly="true" using="MEMBER_OF"> <Property label="Item" name="ItemID" subType="" type="device" typeLabel="Device" typeLabels="Devices"/> <Value displayValue="/All Groups/Defined Tenants/CustomerA/Inventory">***INV_ID***</Value> </Compare> <Compare readOnly="false" using="BETWEEN"> <Property label="Address" name="Address" subType="" type="device" typeLabel="Device" typeLabels="Devices"/> <ValueList> <Value>***Lower IP Address***</Value> <Value>***Upper IP Address***</Value> </ValueList> </Compare> </Match> </Rule> </Rules> </GroupTree>	Rule criteria such as IP Addresses
Optional: create group with same characteristics	As above, replace only the endpoint in the GroupTree element (*Groups* instead of *Collections*) <GroupTree path="/All Groups/Defined Tenants/***TenantName***/Groups">	
Re-Sync DA with PC	(before the corresponding DA IDs can be retrieved, a sync is required, await automatic sync or force it explicitly) PUT http://pc_host:8181/pc/center/webservice/datasources/dataSourceId/**DS_ID_DA**/false Basic Auth User: *global admin* Password: (global admin password) Body: <emptybody/>	Input: DS_ID_DA
Get SNMP profile ID in DA	POST http://da_host:8581/rest/profiles/snmpv1/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <CommunicationProfile.ProfileName type="EQUAL">***ProfileName***</CommunicationProfile.ProfileName> </And></Filter> <Select use="exclude"> </Select> </FilterSelect> Response: ID -> SNMP_ID_DA	Input: ProfileName Output: SNMP_ID_DA
Create Discovery Profile	(example) POST http://da_host:8581/rest/tenant/**TN_ID_DA**/discoveryprofiles <DiscoveryProfile version="1.0.0"> <ActivationStatus>true</ActivationStatus> <SNMPProfileIDList> <SNMPProfileID>***SNMP_ID_DA***</SNMPProfileID> </SNMPProfileIDList>	Input: TN_ID_DA, SNMP_ID_DA, TN_IP_ID, IP Address range (or Host name

	<UseListOfSnmpProfiles>true</UseListOfSnmpProfiles> <IPRangesList> <IPRanges>***IP address range***</IPRanges> </IPRangesList> <HostNamesList> </HostNamesList> <IPListList> </IPListList> <RunStatus>START</RunStatus> <Item version="1.0.0"> <Name>***ProfileName***</Name> </Item> <IPDomainMember version="1.0.0"> <IPDomainID>***TN_IP_ID***</IPDomainID> </IPDomainMember> <DeviceNameRankingList> <DeviceNameRanking>{http://im.ca.com/inventory} ManageableDevice.SystemName</DeviceNameRanking> <DeviceNameRanking>{http://im.ca.com/inventory} Device.HostName</DeviceNameRanking> <DeviceNameRanking>{http://im.ca.com/inventory} Device.PrimaryIPAddress</DeviceNameRanking> </DeviceNameRankingList> <IcmpDiscoveryEnabled>true</IcmpDiscoveryEnabled> </DiscoveryProfile>	list, or IP address list), ProfileName
Get collection DA ID	POST http://da_host:8581/rest/groups/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <Item.Name type="EQUAL">***CollectionName***</Item.Name> </And></Filter> <Select use="exclude"> </Select> </FilterSelect> Response: ID -> COLL_ID_DA	Input: CollectionName Output: COLL_ID_DA
Monitoring Profile	Monitoring profiles are global in the Default Tenant and can be used in other tenants by associating tenant specific collections	
Get monitoring profile DA ID	POST http://da_host:8581/rest/monitoringprofiles/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <Item.Name type="EQUAL">***MPName***</Item.Name> </And></Filter> <Select use="exclude"> </Select> </FilterSelect> Response: ID -> MP_ID_DA	Input: MPName Output: MP_ID_DA
Associate collection with monitoring profile	PUT http://da_host:8581/rest/monitoringprofiles/**MP_ID_DA**/relatesto/groups/**COLL_ID_DA**	Input: MP_ID_DA COLL_ID_DA
Verifications		

Number of devices	GET http://da_host:8581/odata/api/devices/$count?&$filter=((groups/Name eq '{**CollectionName**}')) Basic Auth User: *TenantName_admin* Password: (tenant admin password)	Input: CollectionName
Polled Item Count for test device	GET http://da_host:8581/odata/api/devices?$format=text/csv&$select=ID,Name,PrimaryIPAddress&$filter=((PrimaryIPAddress eq **'IPAddress'**)) Basic Auth User: *TenantName_admin* Password: (tenant admin password)	Input: IPAddress of test device
Supported metric families for test device	http://da_host:8581/odata/api/metricfamilyhistories?$expand=device&$select=VendorCertDisplayName,LastDiscoveryTime,MetricFamilyID,device/Name&$filter=((Status eq 'SUPPORTED') and (device/PrimaryIPAddress eq **'IPAddress'**)) Basic Auth User: *TenantName_admin* Password: (tenant admin password)	Input: IPAddress of test device

Define Roles and Users in tenant		
Roles and Users can be set up by the global admin. This section describes the following aspects 1. get the role IDs for out of the box role definitions in tenant 2. copy custom role from Default Tenant to tenant 3. create user in tenant with given role ID 4. assign default context group to user		
Input parameters	Input parameters for this section: Tenant name Role name Target group User name	
Get standard role ID	NOTE: substitute any blank character in TenantName or RoleName by %20 GET http://pc_host:8181/pc/center/webservice/roles/roleName/**RoleName**/tenant/tenantName/**TenantName**/en-US Basic Auth User: *global admin* Password: (global admin password) Body -> roleID value -> ROLE_ID	Input: TenantName RoleName Output: ROLE_ID
Copy custom role to tenant / get role definition	NOTE: substitute any blank character in RoleName by %20 GET http://pc_host:8181/pc/center/webservice/roles/roleName/**RoleName**/tenant/tenantName/Default%20Tenant/en-US Basic Auth User: *global admin* Password: (global admin password) Response -> copy XML	Input: RoleName Output: XML body
Create role	POST http://pc_host:8181/pc/center/webservice/roles/tenant/tenantName/**TenantName** Basic Auth User: *global admin* Password: (global admin password) Body:	Input: TenantName, XML body Output: ROLE_ID

	(XML copy from previous step) Response: ROLE_ID	
Create user	POST http://pc_host:8181/pc/center/webservice/users/tenant/tenantName/**TenantName**/role/roleId/**ROLE_ID** Basic Auth User: *global admin* Password: (global admin password) Body: <user> <name>**UserName**</name> <description>**UserName** in **TenantName**</description> <enabled>true</enabled> <removable>true</removable> <timezone>Europe/Berlin</timezone> <culture>en-US</culture> </user> Response:	Input: TenantName, ROLE_ID, UserName
Get group ID	GET http://pc_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FDefined%20Tenants%2F**TenantName** Basic Auth User: *global admin* Password: (global admin password) From response, get id value for name="**TargetGroup**" → GROUP_ID	Input: TenantName, TargetGroup Output: GROUP_ID
Assign group	PUT http://pc_host:8181/pc/center/webservice/users/userName/**UserName**/groups Basic Auth User: *global admin* Password: (global admin password) Body: <groups> <group ID="**GROUP_ID**" Name="**TargetGroup**"/> </groups> Response:	Input: UserName, GROUP_ID

Define Threshold profiles in tenant	
Threshold profiles and rules can be created specific to a tenant. An existing profile in the Default Tenant can be chosen as the model for replication. 1. Get DA tenant ID for given Tenant - TN_ID_DA (same as in basic section above) 2. Get DA tenant admin user ID to become owner of the profile (e.g. Tenant_admin user) 3. Create TP folder in tenant 4. Identify a profile (identified by ProfileName) from Default Tenant and use that as a model 5. Create new event rules from model profile in tenant 6. Create new profile in tenant as NewProfileName 7. Associate new profile with TP folder in tenant 8. Associate new profile with relevant group in tenant (identified as GroupName)	
Input parameters	Input parameters for this section: Tenant name Tenant admin user name Folder name Model profile name Target group

Get tenant DA ID	(copied from above base tenant setup section) POST http://da_host:8581/rest/tenants/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter><And> <Tenant.Name type="EQUAL">***TenantName***</Tenant.Name> </And></Filter> <Select use="exclude"></Select> </FilterSelect> Response: <ID> → TN_ID_DA	Input: TenantName Output: TN_ID_DA
Get tenant admin DA userID	POST http://da_host:8581/rest/users/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter> <And> <User.Name type="EQUAL">**AdminName**</User.Name> </And> </Filter> <Select use="exclude"> <User use="exclude"> <UserID use="include"/> </User> </Select> </FilterSelect> Response: <UserID> → ADMIN_ID_DA	Input: AdminName Output: ADMIN_ID_DA
Create TP folder	POST http://da_host:8581/rest/tenant/**TN_ID_DA**/eventprofilegroups Body: <EventProfileGroup version="0.0.0"> <Item version="1.0.0"> <Name>**FolderName**</Name> <Description>scoped profiles folder</Description> </Item> </EventProfileGroup> Response: ID -> FOLDER_ID	Input: TN_ID_DA FolderName Output: FOLDER_ID
Get model profile ID	POST http://da_host:8581/rest/eventprofiles/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter> <And> <Item.Name type="EQUAL">**ProfileName**</Item.Name> </And> </Filter> <Select use="exclude"> </Select> </FilterSelect> Response: ID -> PROF_ID	Input: ProfileName Output: PROF_ID
Get the profile content	GET http://da_host:8581/rest/eventprofiles/**PROF_ID** Response: get the IDs in <EventRuleIDList> → RULE_ID	Input: PROF_ID Output: RULE_ID

Get event rule definition	GET http://da_host:8581/rest/eventrules/**RULE_ID** Response: Remove the following lines from the XML body: <ID>…</ID> <CreateTime>…</CreateTime> copy XML	Input: RULE_ID Output: XML body
Create event rule	POST http://da_host:8581/rest/eventrules Body: XML copy Response: <success> value -> NEWRULE_ID	Input: XML body Output: NEWRULE_ID
Create profile	POST http://da_host:8581/rest/eventprofiles Body: <EventProfile version="2.0.0"> <OwnerID>**ADMIN_ID_DA**</OwnerID> <LastModifiedUserID>**ADMIN_ID_DA**</LastModifiedUserID> <LastModifiedTime>**TIMESTAMP**</LastModifiedTime> <EventRuleIDList> <EventRuleID>**NEWRULE_ID**</EventRuleID> </EventRuleIDList> <Enabled>true</Enabled> <Item version="1.0.0"> <Name>New**ProfileName**</Name> <Description/> </Item> </EventProfile> Response: <success> value -> NEWPROF_ID	Input: ADMIN_ID_DA TIMESTAMP NEWRULE_ID NewProfileName Output: NEWPROF_ID
Associate new profile with folder	PUT http://da_host:8581/rest/eventprofiles/**NEWPROF_ID**/relatesto/eventprofilegroups/**FOLDER_ID** Body: <emptybody/> Response:	Input: NEWPROF_ID FOLDER_ID
Get group DA ID	POST http://da_host:8581/rest/groups/filtered Body: <FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="filter.xsd"> <Filter> <And> <Item.Name type="EQUAL">**GroupName** </Item.Name> </And> </Filter> <Select use="exclude"> <Item use="include"> </Item> </Select> </FilterSelect> Response: ID -> GROUP_ID_DA	Input: GroupName Output: GROUP_ID_DA
Associate new profile with group	PUT http://da_host:8581/rest/eventprofiles/**NEWPROF_ID**/relatesto/groups/**GROUP_ID_DA** Body: <emptybody/>	Input: NEWPROF_ID GROUP_ID_DA

	Response:	

Share customizations in Default Tenant		
Share custom dashboard		
Input parameters	Input parameters for this section: n/a	
Identify page ID	In CA PC, open the desired dashboard From the browser URL, get the page ID from the &pg= parameter → PG_ID	Output: PG_ID
Export dashboard xml by global admin	GET http://pc_host:8181/pc/center/webservice/dashboards/**PG_ID** Basic Auth User: *global admin* Password: (global admin password) Body -> save	Input: PG_ID
Edit dashboard xml	(optional) Verify the following xml tags are correctly set for the tenant environment, or adjust accordingly <dashboardMenu> <menuItem> <dashboardTitle> <menuTitle>	
Import dashboard xml in tenant by tenant admin	POST http://pc_host:8181/pc/center/webservice/dashboards/import Basic Auth User: *TenantName_admin* Password: (tenant admin password) Body: xml from previous step	
Share context page customizations		
Custom tabs that are created on device or device component context page in the Default Tenant can be copied to other tenants using a command line utility. Please contact your Engineering Services Architect for more information.		