Application Dependency Mapping (ADM) Guide

Change	User	Date
Initial Doc	Daniel Holroyd	Dec 2023
Format and content changes	Stan Tusinski	Mar 2024
Additional content added, recommended steps and Troubleshooting	Stan Tusinski, James Scrimale	Apr 2024
Final Draft completed	Stan Tusinski	May 2024
Updated with 19.05 changes including Application Groups, Business Services and Calculation Logic	Stan Tusinski	Mar 2025
Updated based on feedback from colleague review	Stan Tusinski	Oct 2025

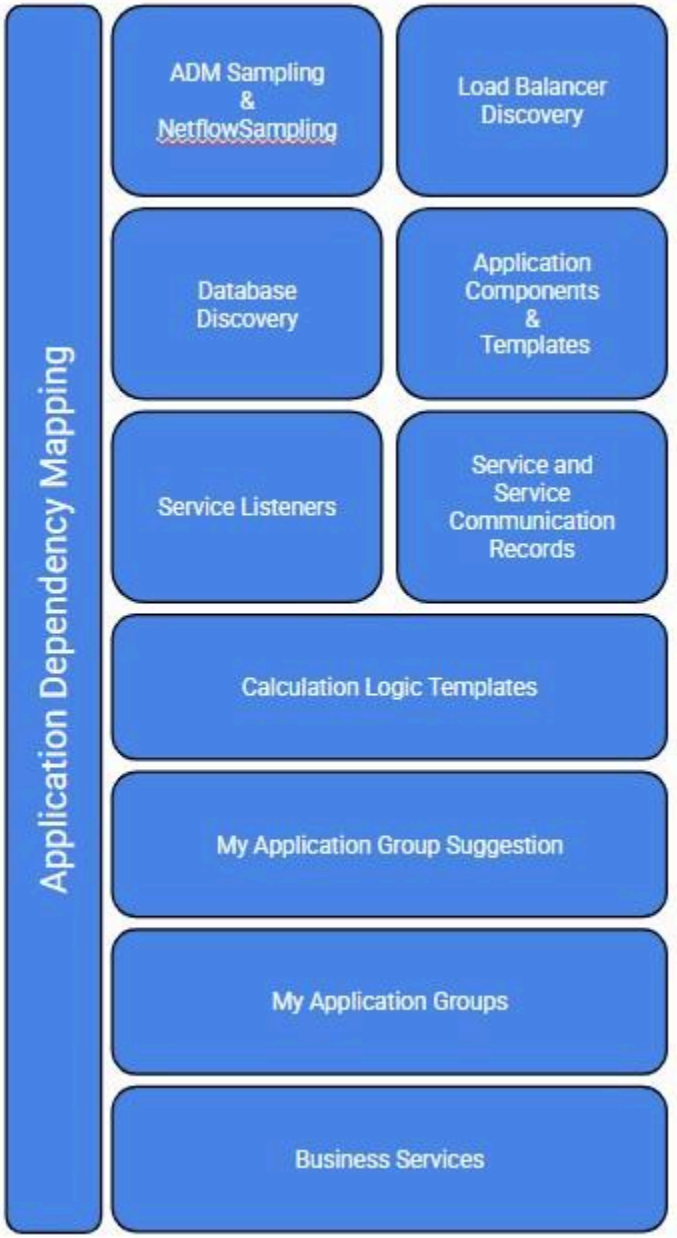
Application Dependency Mapping (ADM) is a separate feature licensed within Device42. Netflow which is also mentioned in this guide is also a separate licensed feature within Device42 and can add additional connection information from the network devices to your Business Services.

This guide will outline the steps needed to be successful with ADM. There is a particular flow of actions that we recommend as best practices to get the best results out of ADM. The diagram on the right showcases the sections within the tool that make up ADM as a complete solution and are explained in detail throughout this document. The work flow is top to bottom.

ADM functionality at its core is based on Service to Service communications.

The communication information is collected via Netstat in most instances which allows Device42 to build up relationships between servers over time. These communications are represented as the “Listener” and “Client” when viewing the raw data in Device42.

The following topics will walk you through how Application Groups are configured within your environment.

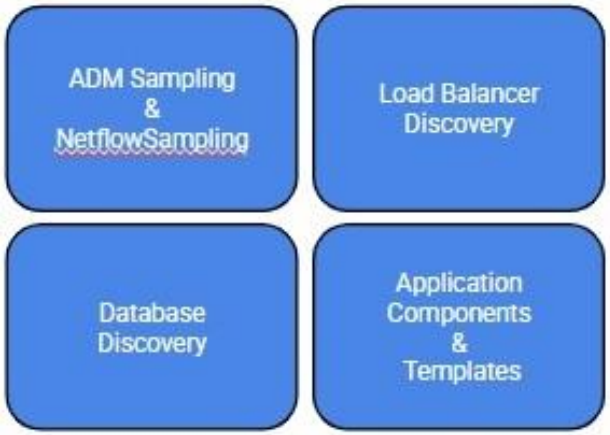


Discovery

The first four bullets listed below are the discovery of foundational information that will be used to build out the relationships. This is made up of four main categories.

- ADM & Netflow Sampling
- Load Balancer Discovery
- Database Discovery
- Application Component and Templates

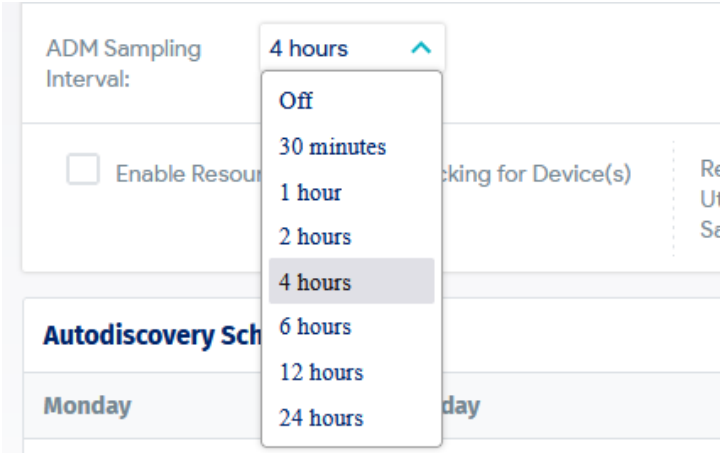
These four categories will build up the underlying dataset used to map applications



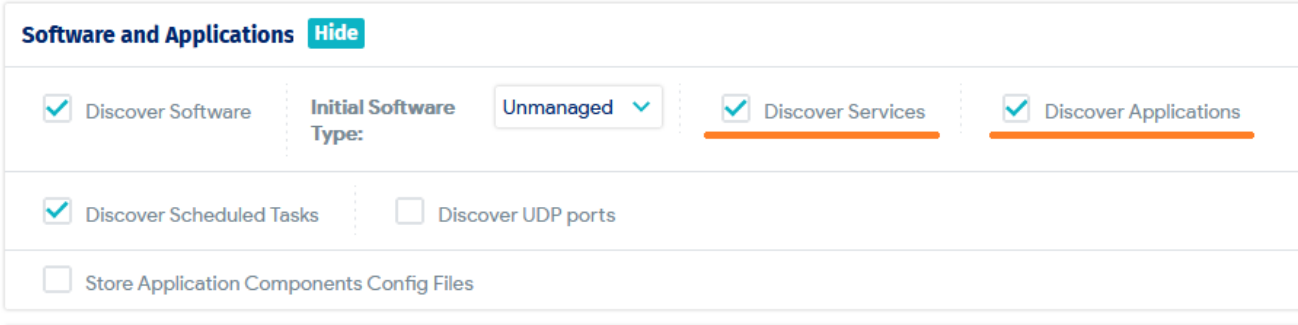
ADM & Netflow Sampling

Identify the application servers that will be added to the Discovery Job which will have the option for ADM sampling enabled.
The supported platforms are Windows and *Nix based servers.
To enable ADM scanning you will need to enable “ADM Sampling Interval:” on the Windows/*Nix job.
The default setting is 4 Hours, if you are interested in gathering information faster where connections aren’t persistent this can be adjusted to sample more frequently in the beginning. After some time has passed it’s best to adjust it back to the default.
This will pull a list of server communication at that interval.

***NOTE: in some cases more frequent sampling is required if performing a migration project.**



If you don't see the option of "ADM Sampling Interval:" Ensure you have the "Enterprise Application Discovery" and "Services Discovery" modules with your license and check that you have set the platform at the top of the page to Windows or *Nix.
Scroll down within the Discovery Job Configuration and under the "Software and Applications" section be sure you have "Discover Services" and "Discover Applications" ticked.



The next time the Discovery job runs it will discover the Services and Applications on the systems.
When ADM sampling is enabled it will create child jobs from the Windows/*Nix job you just created. These are called **Periodic Jobs** and are linked to each of the systems that were discovered in the Discovery job. Be sure to keep the original Discovery Job that was used as it's linked to the Periodic Job that was created so any adjustments to the ADM Sample Frequency is managed by the Discovery Job.
More information can be found here on them.

Netflow is an optional module for Application Groups, it allows you to configure the Remote Collector to receive Netflow data from routers and switches. This is useful if the connections that make up your application are ephemeral and might not be captured in the available time configured in the "ADM Sampling Interval:" Details how to configure Netflow can be found here.

Load Balancer Discovery

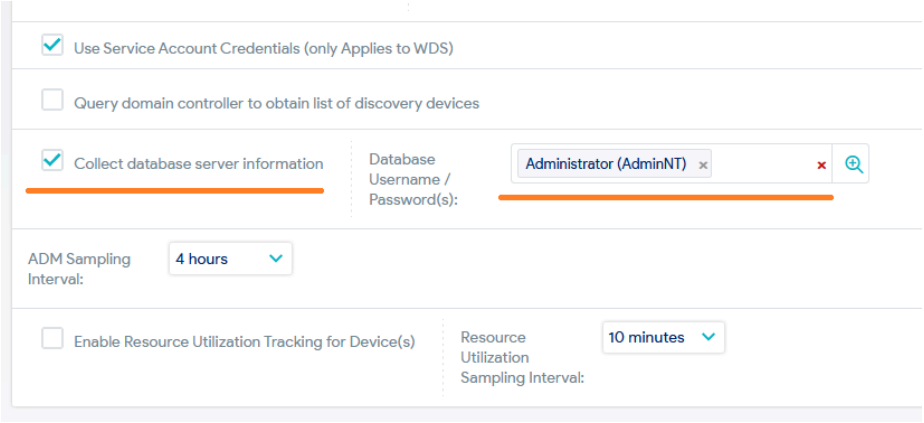
SNMP will get you the device information along with the virtual servers from the Load Balancers.

Using the UCS/Load Balancer API discovery gathers the netstat data from F5 load balancers to correlate VIP data with backend pool members. These scans will provide additional data for ADM Discovery Jobs. Device42 recommends this for additional Application Groups discovery data.
The order is to create the SNMP scan first to capture the Device information then add the UCS/ACI Discovery job which will gather the resource information via the API and complement the Device discovery. This additional data can also be seen when using the Application Dependency Mapping (ADM) module.

Database Discovery

Due to the nature of how applications are architectured a Database will more than likely make up part of your application. It is also important if you are trying to separate your application mapping where there are shared Database instances involved. For example if a Database server is hosting multiple applications mapping it from the server level will result in all the communications from that server being grouped together, you will want to separate this and start the mapping from the Database itself (This will be covered in Application Group Calculation Rules filters later in the guide).

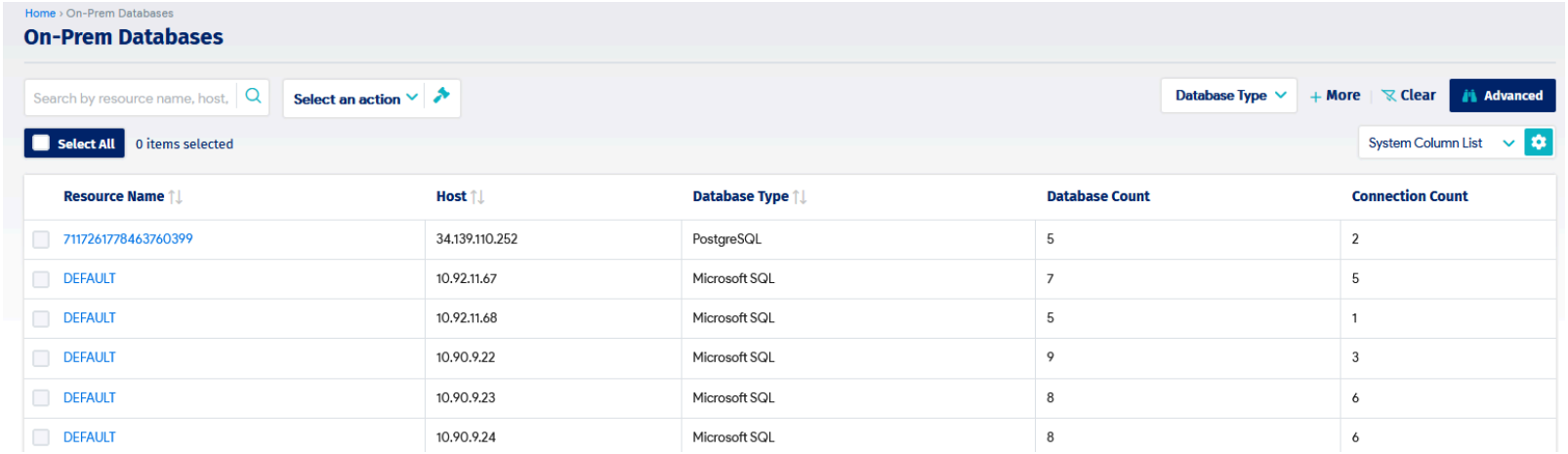
Enabling Database discovery is similar to enabling ADM sampling, It is done from the Windows/*Nix job.



Similar to ADM sampling if you don't see the option for Collect Database Server information, ensure you have the Enterprise Application Discovery and Services Discovery modules with your license and check that you have set the platform at the top of the page to Windows or *Nix.
Scroll down within the Discovery Job Configuration and under the "Software and Applications" section be sure you have "Discover Services" and "Discover Applications" ticked.

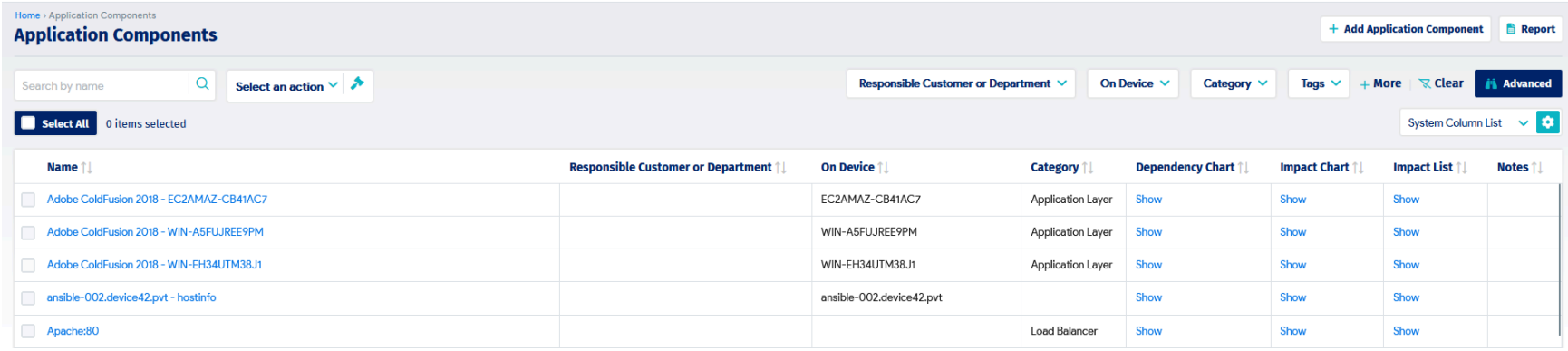
Information on how to configure Database discovery can be found here.

The desired outcome of Database discovery is to have your Database listed in the On-Prem Databases page, with values in the “Database Count” and “Connection Count” columns. If you see 0 in the “Connection Count” this means we have not been able to fully discover the database, permission levels of the account are normally the cause of this.



Application Components and Templates

Application Components are part of what will build the application relationships. These are created during the deeper Advanced Discovery phase of the Operating System level Windows/*Nix Discovery job run. Device42 will automatically create these components of the application based from a predefined list within the product. Device42 uses most commonly used Application Components in three main Categories, Database, Web Server and Application Layer that are used in application design.
These are created by looking at the service name. You can view the Application Components that Device42 has detected by going to Applications > Application Components in the UI.

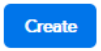


This is the list of writing this document, an up to date list can be found here.

Application	Services Discovered	Configuration Information Imported
Apache	Yes	Web site titles, bindings, installed apps (ie, Drupal, Wordpress, SugarCRM)
Coldfusion	Yes	Web servers, data sources, jrun
DB2	Yes	Version, install path, products, instances, port, protocols, config files *.cfg
Hadoop	Yes	Version, home directory, policies, configurations, config files, nodes, properties, resource manager
IIS6	Yes	Web site titles, bindings, installed apps (ie, Drupal, Wordpress, SugarCRM)
IIS7	Yes	Web site titles, bindings, installed apps (ie, Drupal, Wordpress, SugarCRM)
MariaDB	Yes	Config file path, protocols/listening ports, data paths, config files, version

Microsoft SQL Server	Yes	Instances, shared memory, TCP/IP, named pipes, data paths
MongoDB	Yes	Config file paths, protocols, data paths
MySQL	Yes	Config file path, protocols/listening ports, data paths
Oracle Database	Yes	Instances, protocols, data paths
PostgreSQL Database	Yes	Config file paths, protocols, data paths
SAP Hana	Yes	Version, HDB Info, config files, ini files
Sybase	Yes	Instances, data path, config path, config files, base path, data server
Tomcat	Yes	Config files, version, home path, JVM, architecture
WebSphere	Yes	Version, config files, DTD path, product, path, servers

Don’t worry if some of the services you use in your application are not listed above, Leveraging Application Component Templates allows you to account for application components that are not on the list or for in house developed solutions.

Application Component Templates can be created from going to Applications > Application Component Templates in the UI. This will list all the templates you have created. To create a new template click the 

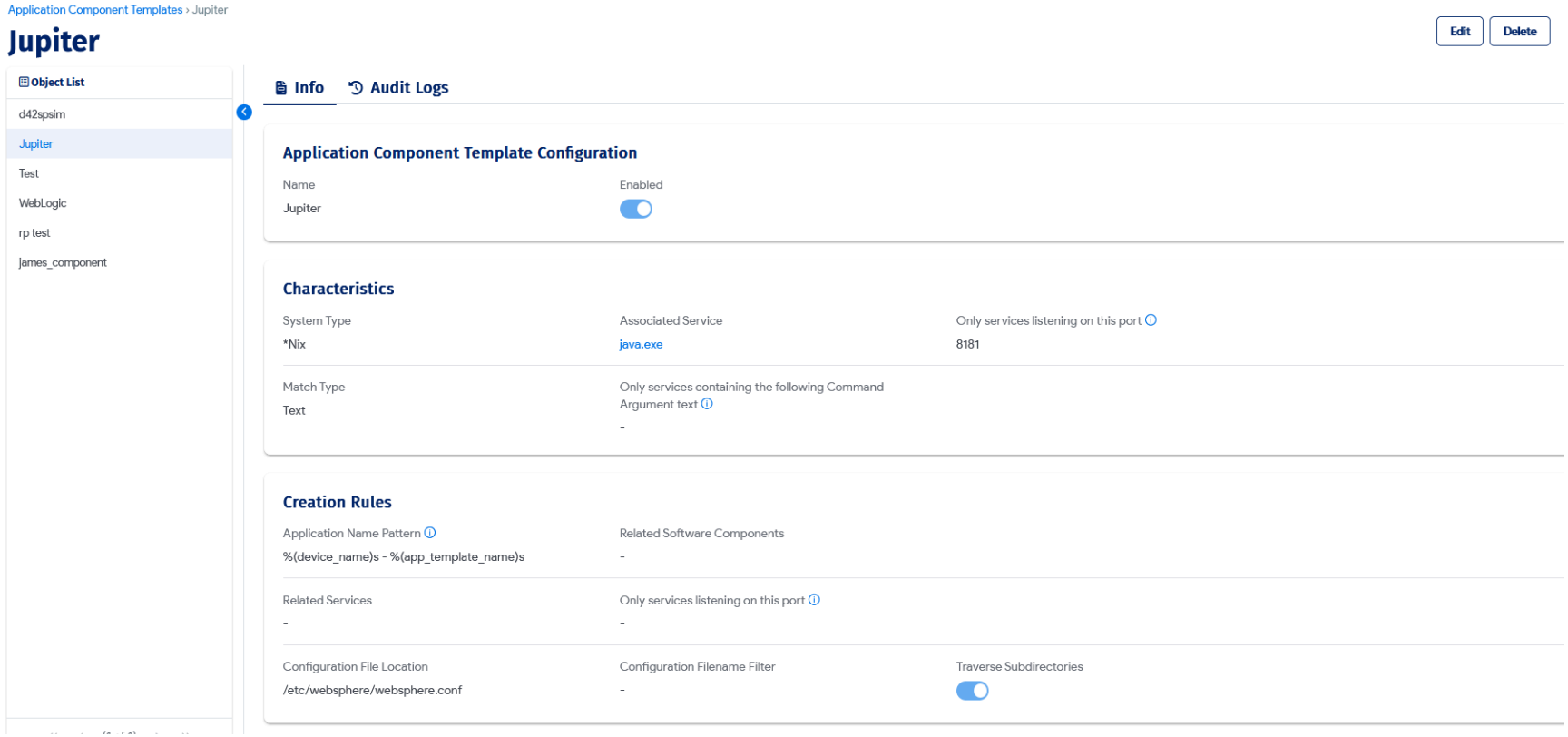
button in the top right hand corner.

On this page you are able to define detection guide rails for the tool so that when it finds services running that match the defined criteria Device42 will create an Application Component for it.

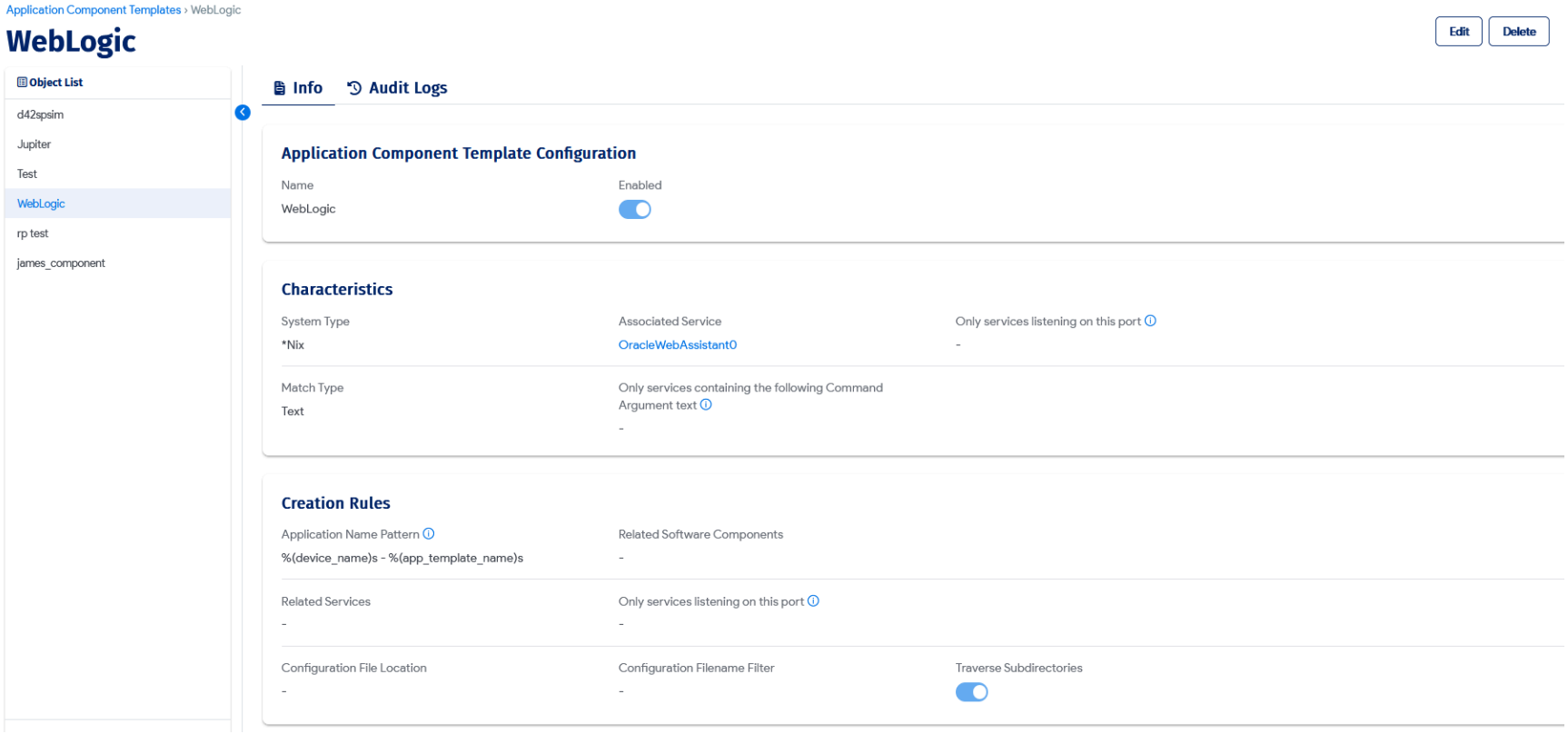
Application Component Templates give a lot of granular control for detecting Application Components to avoid false positives. At its simplest form, All an Application Component Template needs is a System type, Windows or Nix and an Associated Service to start creating Application Components at discovery. This works great if you have a clearly defined service name.

A good example of a where the rest of the controls in the Template creations provide value is if part of your application is made up of a Java service. Java is a very common service you might find throughout your environment, but not all instances of it will be relevant to your application. You can leverage these controls to fine tune which of the Java service instances you create Application Components from. For example you could define the listening port for the Java service, the Command Argument, or other services that must be present on the device or it to create an Application Component.

In the example below a simple Java application instance listening on Port 8181 has been identified and given the name of “Jupiter”. For any Java Service instance listening on port 8181 found during Discovery it will be named “Jupiter” under the Application tab in the Device record.



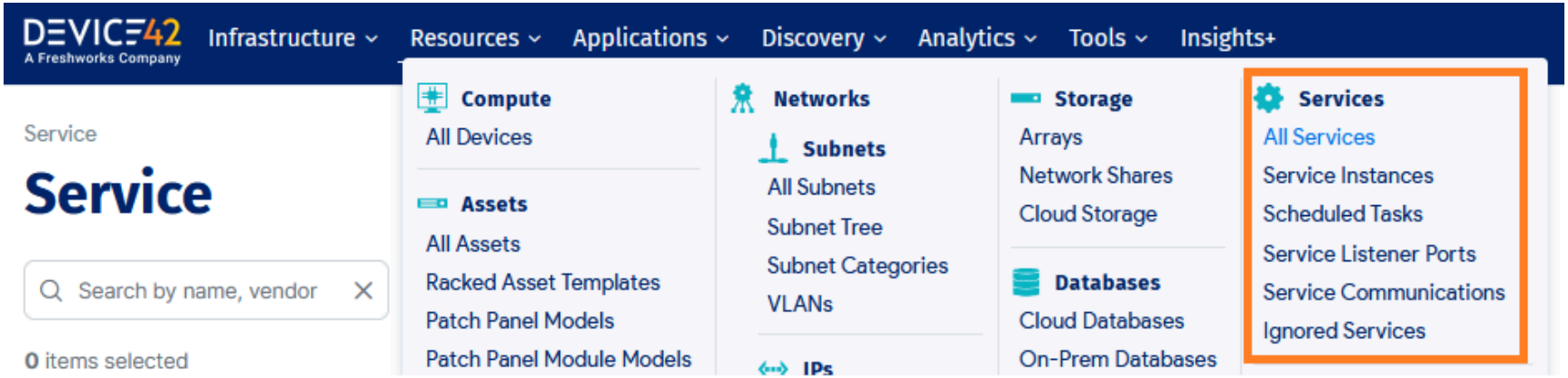
In the example below a more complex application instance has been identified with OracleWebAssitant0 as the Associated Service and a Related Software Component of “IBM WebSphere Application Server V9.0” and given the name of “WebLogic”. For any OracleWebAssitant0 Service instance found during Discovery it will be named “WebLogic” under the Application tab in the Device record.



More information on creating Application Component Templates can be found here.

Service and Service Communication Records

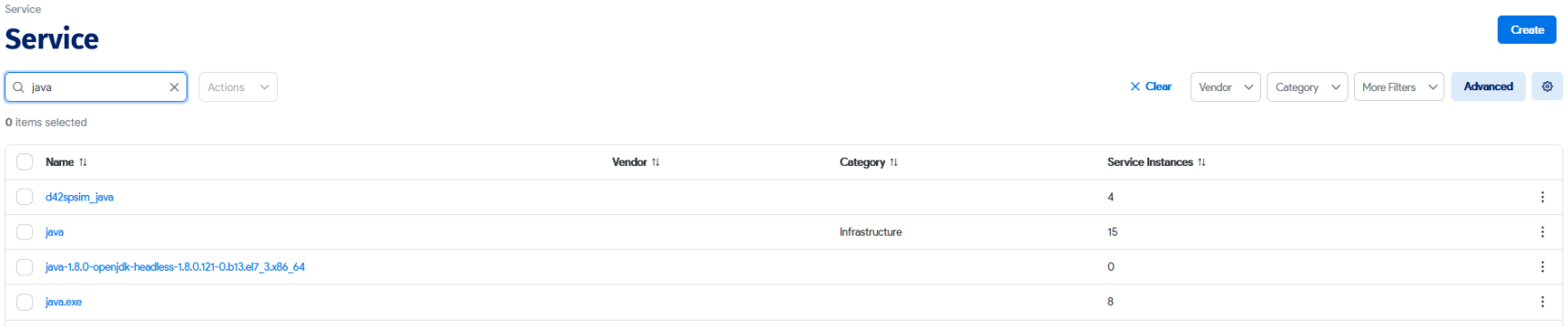
All of the Service Communications we discover as part of the ADM Sampling are stored in the Services section of Device42 under Resources.



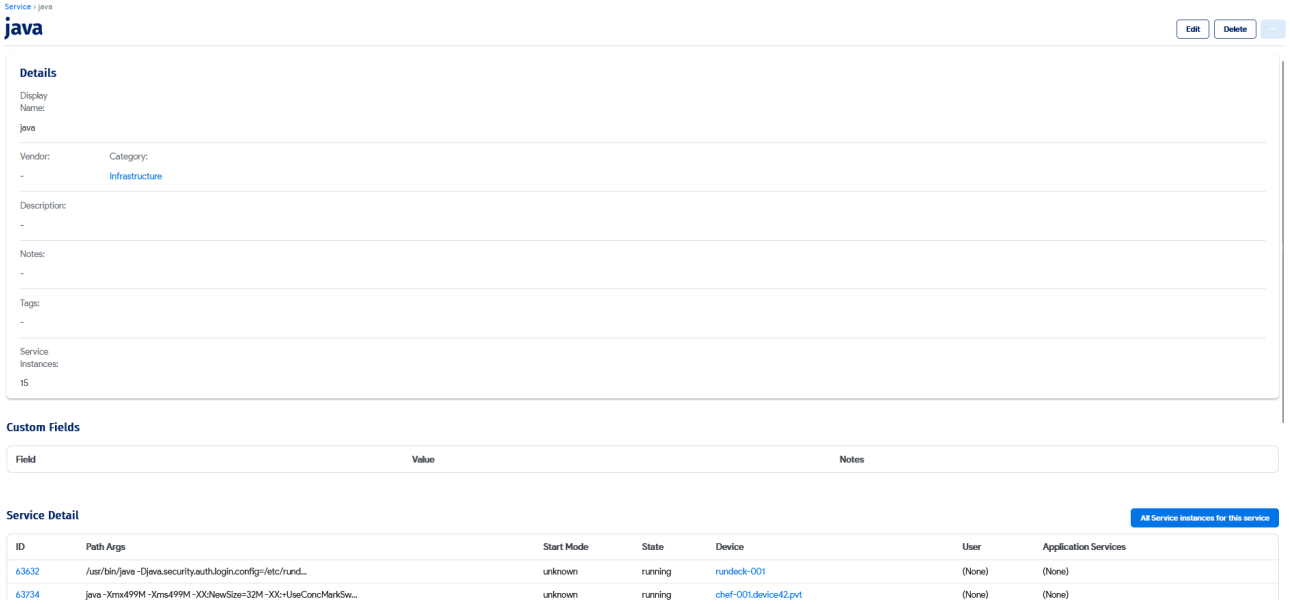
The two most useful pages in this section for ADM are “All Services” and “Service Instances”.
All Services provides a list of each identified Service Name that has been Discovered along with providing the quantity of “Service Instances” that have been Discovered.
Service Instances are the unique instances that are running under that Service name.

A good example is Java, you may find a single Java Service listed with multiple Instances. Drilling into the Java Service will provide a list of all the Instances that are “listening” for a connection.

Below is an image showing Services that contain the word “java” in them, notice the number of instances to the right.



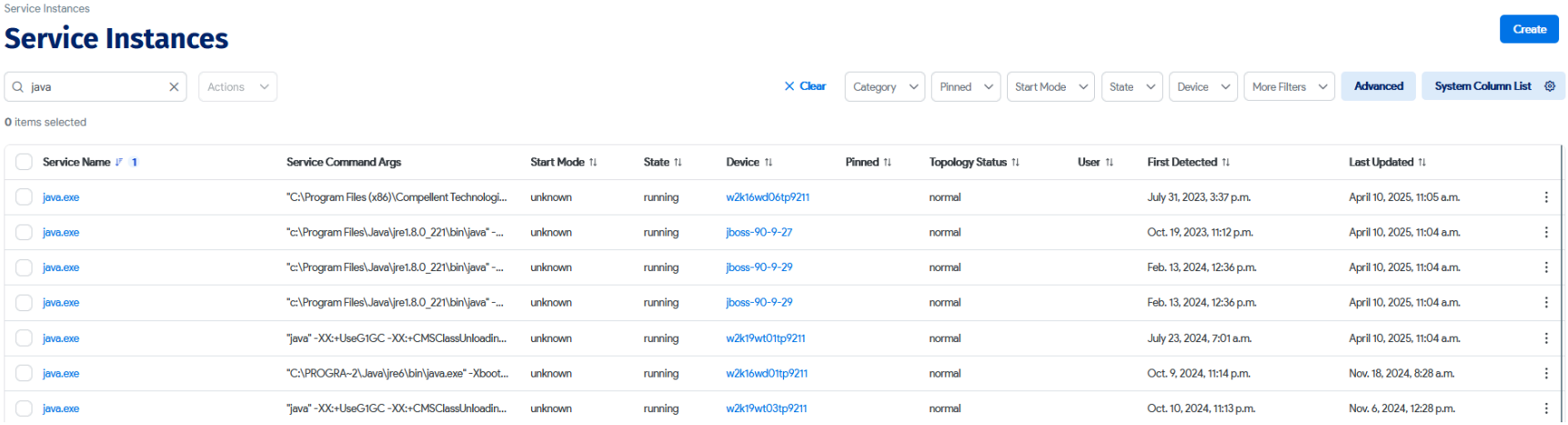
After drilling into the “java” service above we can see a list of the individual instances and the Devices they are running on.



The “All Services” page can be useful when looking to identify noisy services such as agents within your environment. You can filter this page from high to low on the Service Instance count.

Undiscovered Listening Services is nearly always at the top when sorting from high to low. You can learn more about how to manage these here.

Service Instance page lists out all of the services records in your environment, This page is useful when looking for values to be used to create Application Component Templates, such as the “Path Args”.

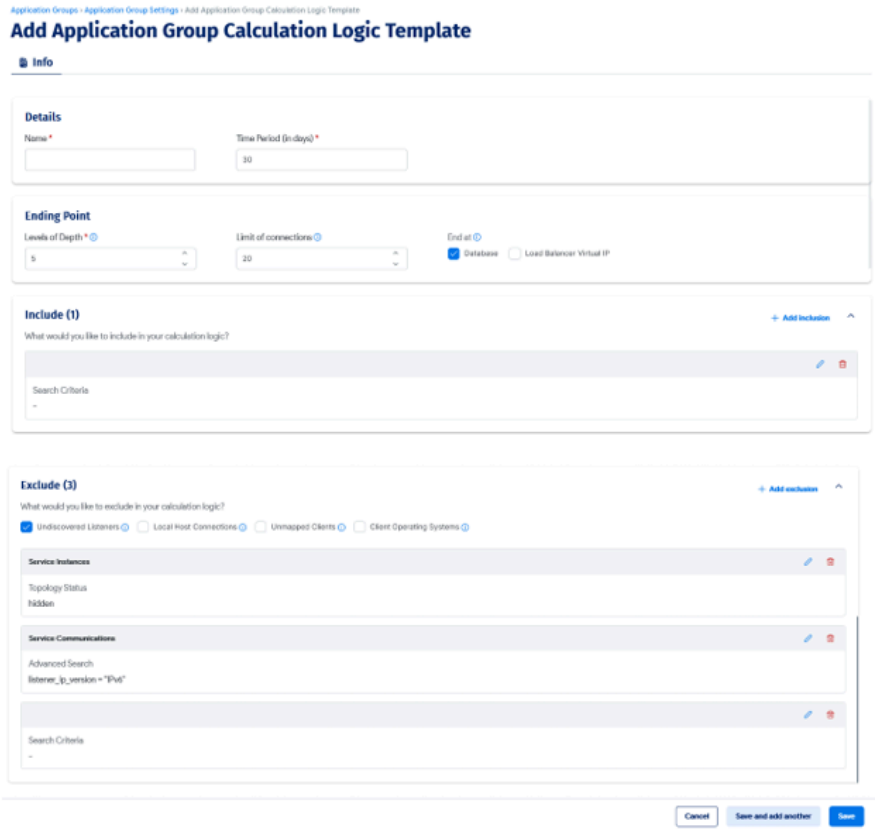


Application Groups

Application Groups are created based on Application Group Calculations configured from the Server to Server Service communications captured during the Discovery process. There are two main components to this, the Application Group Calculation Rules and the Application Group Calculation Logic Template. The Calculation Logic Template can be defined using a Form for a simple way to calculate your Application Group or by using a DOQL (Device42 Object Query Language) database query for more complex calculations. It's recommended to use the Form based Calculation Logic over the DOQL based logic. You can create Calculation Logic tailored to each Business Service in your environment. Once they are created you apply the Calculation Logic rule to the Application Group Calculation Rule for the Business Service you are mapping.

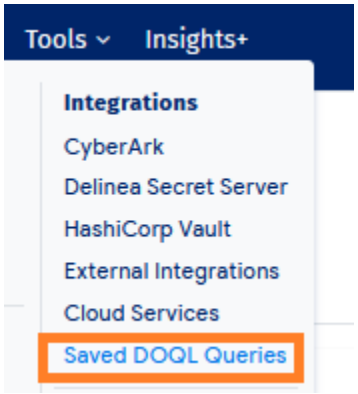
Application Group Calculation using Forms

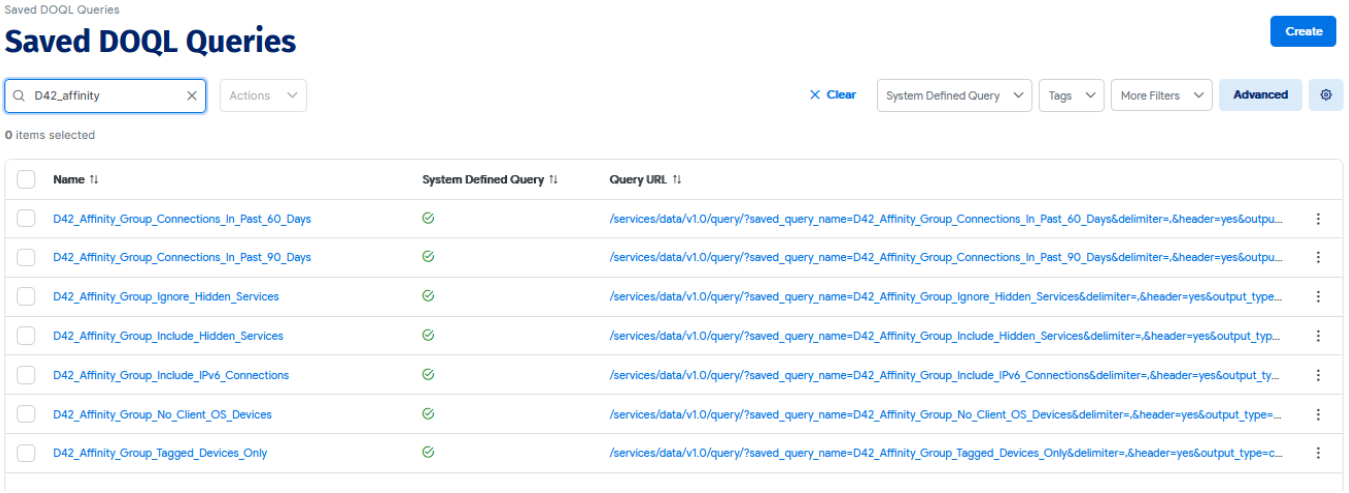
There is a default Form "D42 Default Template" that can be used or you can create your own. Creating an Application Group Calculation Logic Template using a Form provides a wizard like interface where you can select the levels of Depth or connections the Ending Point will have and limit the number of connections made when calculating the listener and client connections. Inclusions are typically left blank unless you need to focus on a particular group of systems. Exclusions allow you to specify Specific Application Components, Devices, Resources, Service Communications or Service Instances that will be excluded from the Calculation Logic Calculation for that specific Application Group. Ie: antivirus software could be excluded.



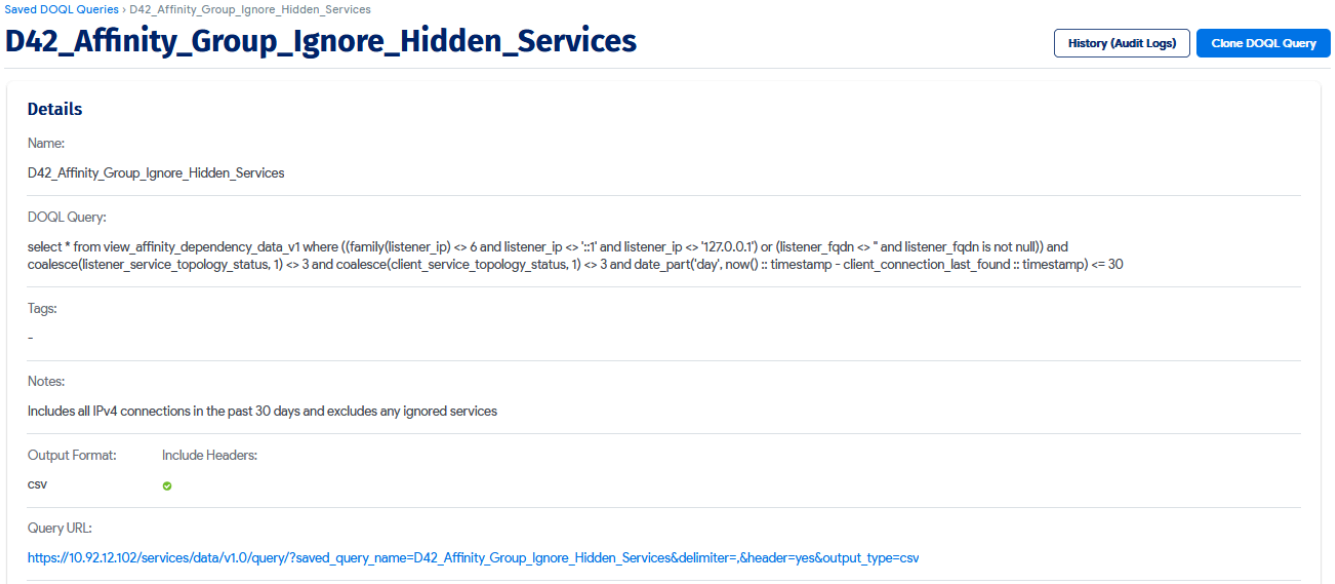
Application Group Calculation Logic using DOQL

There are System Defined DOQL queries that can be used for Application Group Calculation Logic Templates. The most common one used is “DOQL “D42_Affinity_Group_Ignore_Hidden_Services” or you can create your own.
The DOQL controls what data is passed on to Application Groups to calculate the dependencies. You can see what Data is currently being passed through to Application Groups to be calculated by doing the following, go to Tools > Saved DOQL Queries.
Search here for the word “D42_Affinity”, this will help you identify the default DOQL used. You can click on the URL to download a CSV of the raw data that is passed to the Application Groups.





If you click on the Name of the DOQL you will be able to see the query that is being run on all of your data to build out the dataset used by Application Groups. This is where you can do bulk editing to manipulate the dataset.

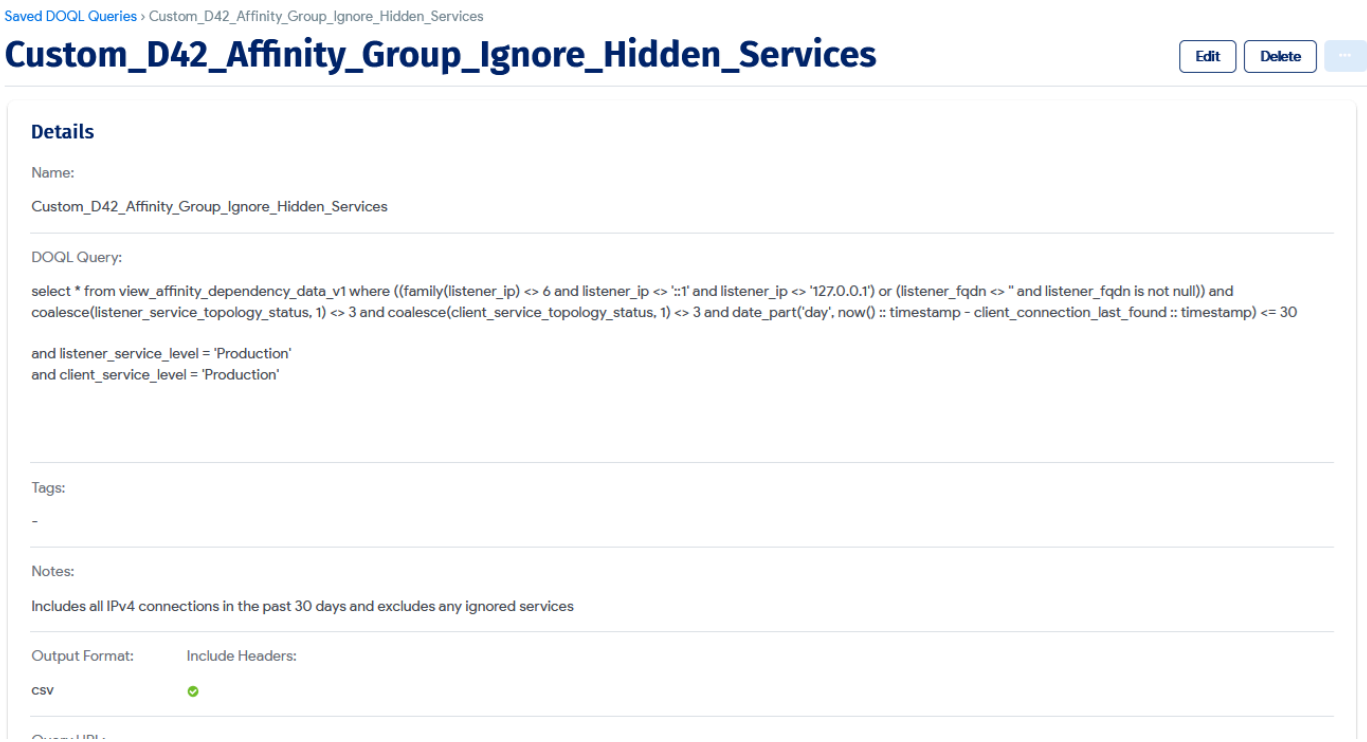


To make changes to this we will need to make a clone of this DOQL as we can not edit system defined ones. Click Clone in the top right hand corner.

Some examples of data manipulation you can do here;

If you have different service levels in your environment and you only want to have Application Groups created for Servers with the service level set as production, we could add two lines.

and coalesce(listener_service_level,”) = 'Production'
and coalesce(client_service_level,”) = 'Production'

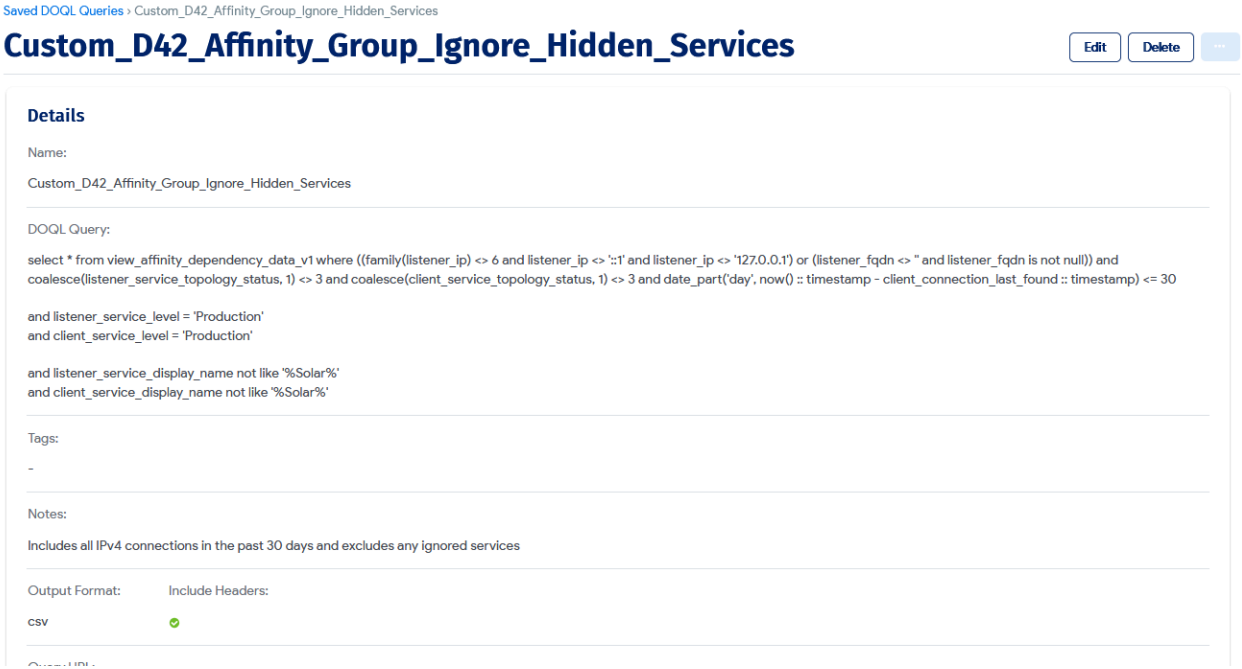


These values were selected after reviewing the CSV document we downloaded earlier. This means that any AGs data we have for services attached to servers that don't have Production set as their service level will not get passed through to the Application Groups.

Another useful modification you can make here is if you are finding your Application Groups to be quite large in size, a common cause of this can be Backup or Monitoring tools, as these function like Applications, which have Databases but work like a mesh network with devices connecting to each other. These are supporting applications and generally are not desirable to map. Utilizing the DOQL to cut these noisy services out can give great results.

and coalesce(listener_service_display_name,'') not like '%Solar%'
and coalesce(client_service_display_name,'') not like '%Solar%'

With these two lines we will exclude all services that include “Solar” the % is a wildcard in DOQL and can be used as a catch-all. In this instance any service display name with “Solar” in the name, whether it has text in front or behind of it will be excluded.



You can always test the impact of these changes by saving it and clicking on the Query URL and checking if the changes had the desired impact on the CSV.

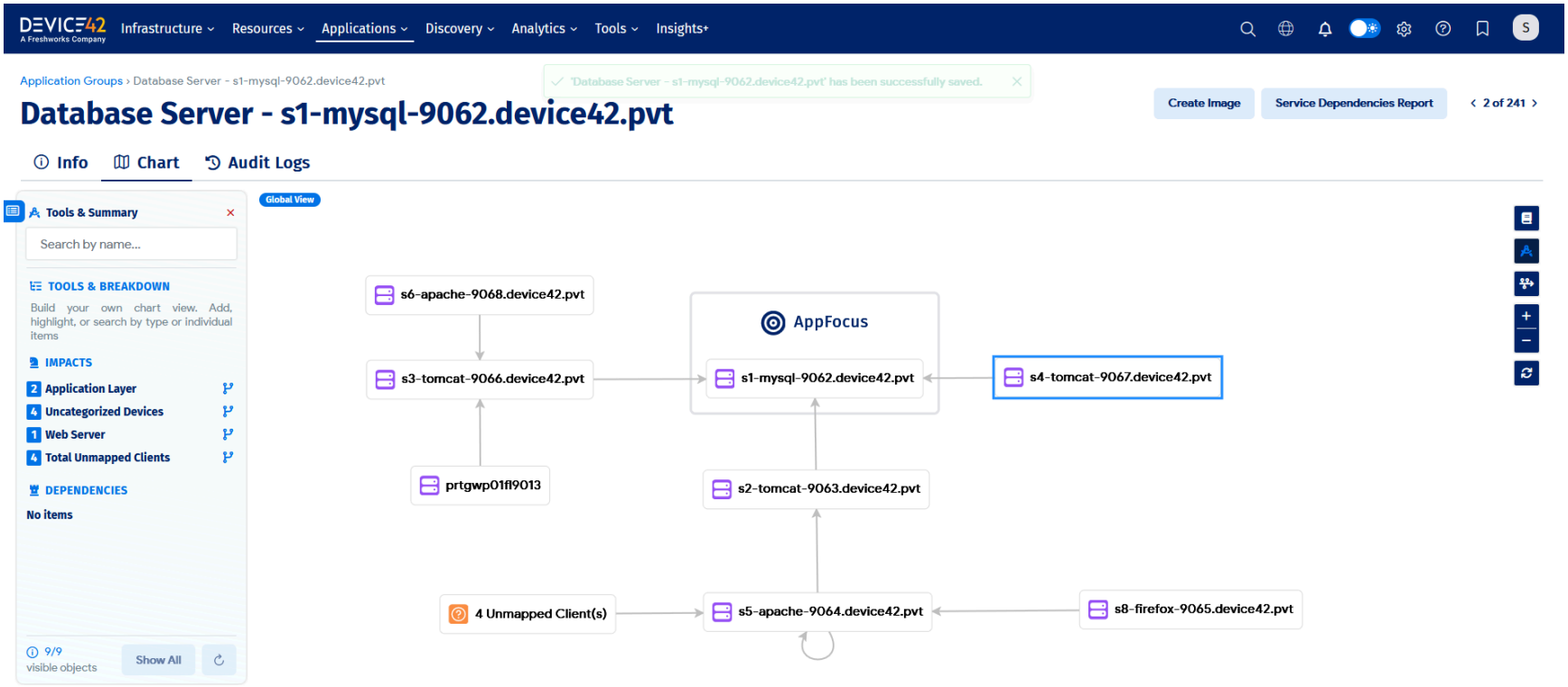
Application Group Calculation Rules

Application Group Calculation Rules are the foundations of building Application Groups, they are used as the starting point of your Application Group.
There are 3 system defined Application Group Calculation Rules.

- Database
- Database Server
- LB VIP

The majority of applications will involve a Database in their architecture, This allows Device42 to leverage the system defined Application Group Calculation Rules to start the mapping. The “Database -” Application Group Calculation Rules give the best control and map connected from the Database level, ideal where you have Database Instances hosting multiple applications. If the application is deployed in such a way where there are dedicated Database Instances per application you can leverage the “Database Server -” Application Group Calculation Rule, this will map connections from the server level and include all connections to the Database instance.

Database Server

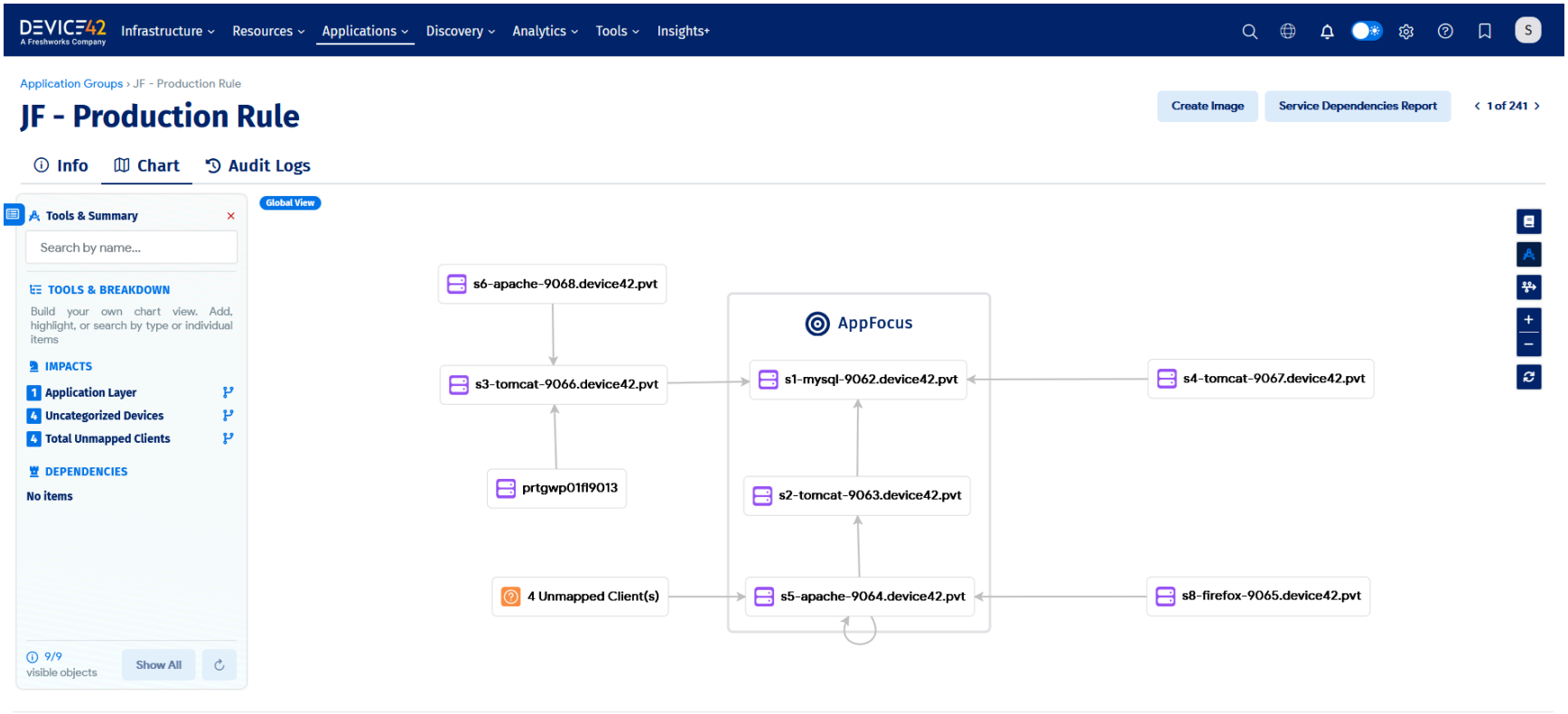


You can see the Database Server, **s1-mysql-9062.device42.pvt** has several systems connected to it, you can assume that each tomcat server is a unique Application Server. In this diagram there is no way to identify each Application uniquely, this is where Application Group Calculation Rules are used.

There are also 4 Unmapped clients accessing the **s5-apache-9064.device42.pvt** Webserver.

An Unmapped Client is a device that has not been Discovered by Device42 and is not part of the discovered inventory.

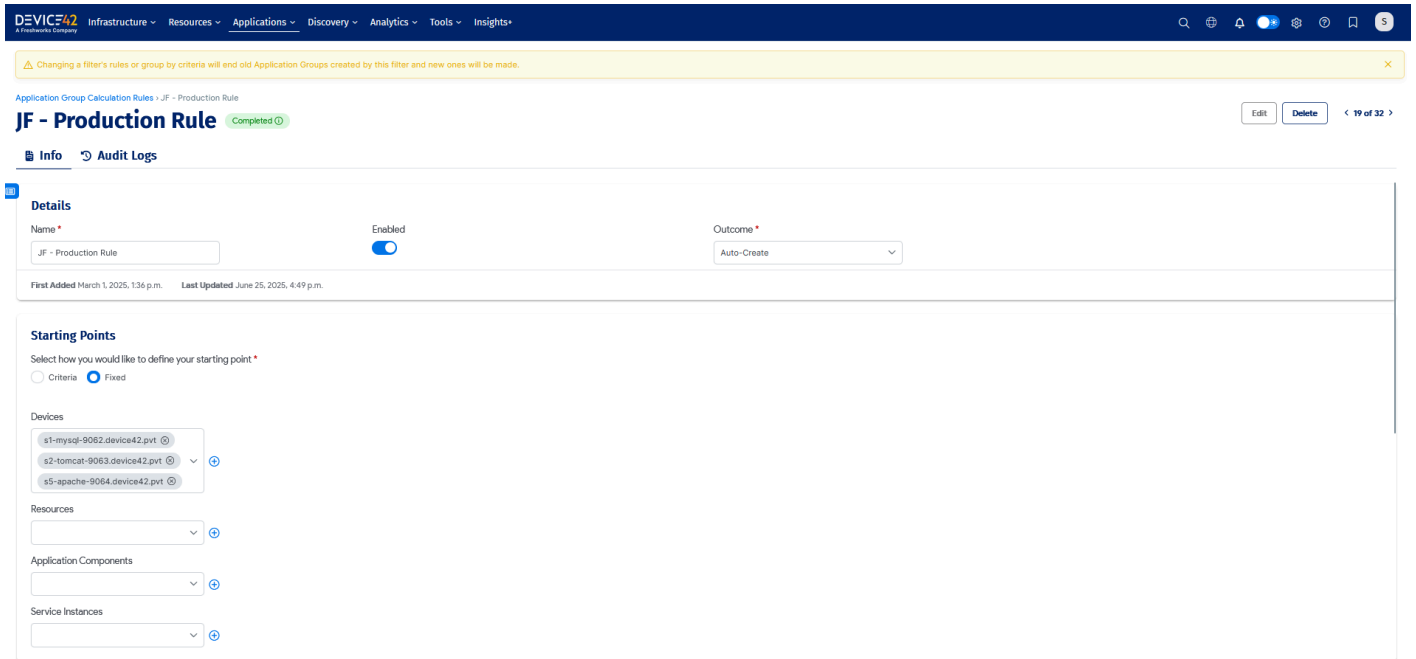
Application Group Calculation Rule example



You can see the Database Server, **s1-mysql-9062.device42.pvt** along with **s2-tomcat-9064.device42.pvt** and **s5-apache-9064.device42.pvt** are in the AppFocus Window. These three systems make up the **JF - Production** Application Group.

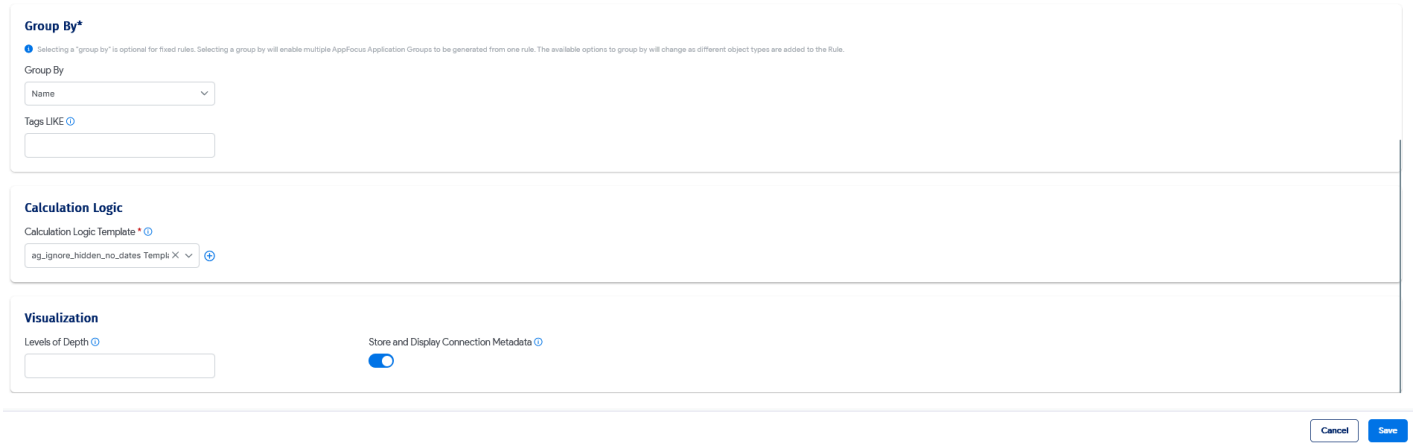
Creating a Custom Application Group Calculation Rule

Application Group Calculation Rules can be created using several methods to define how they will be processed. First you define your Starting Point, this can be search criteria or a fixed set of Application Components, Devices, Resources or Service Instances.



The outcome can be a Suggestion or Auto-Create, next select the Group By option (required when using a Fixed Starting Point) and the Calculation Logic that will be used.

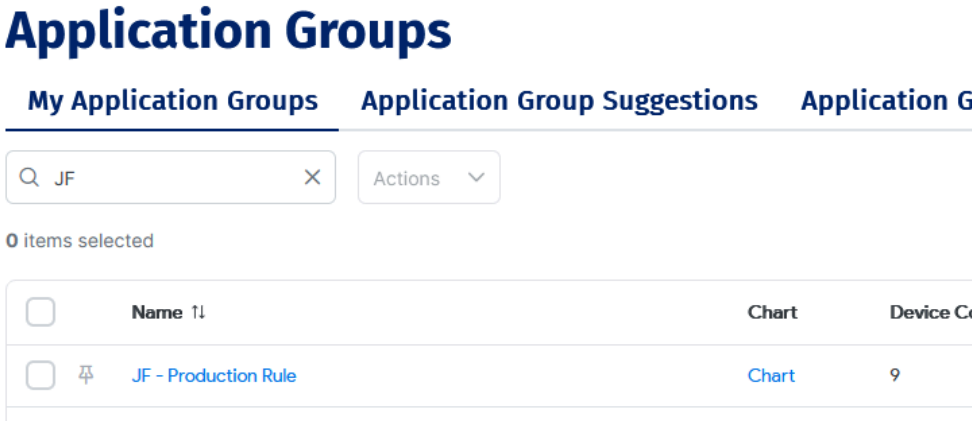
In this configuration the Group By is set to Name, additional options can be selected, one example is **Service Level** where multiple Application Groups would be created each based on the Service Level such as Production, Development and so on.



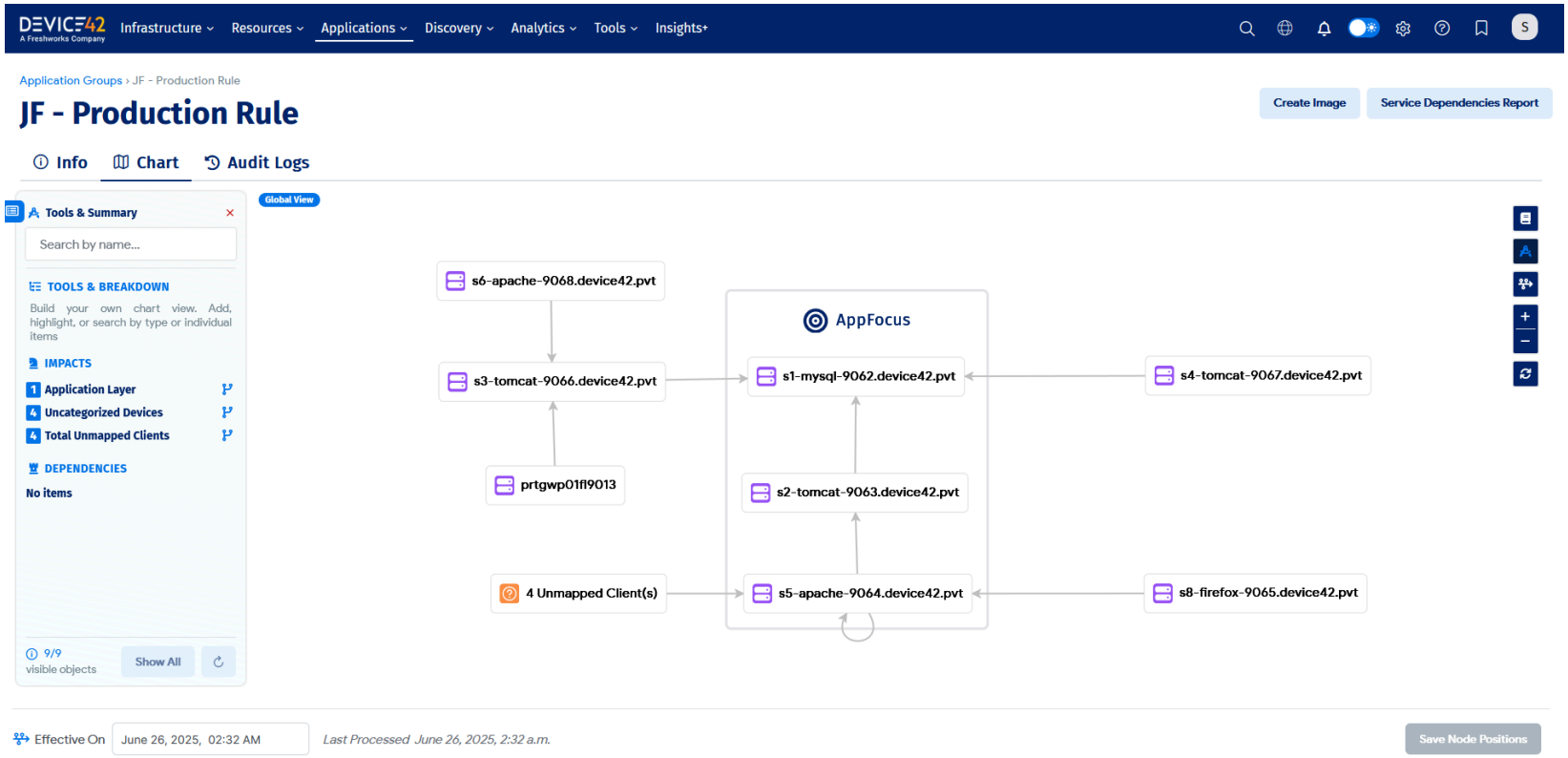
Once you have configured the Application Group Calculation Rule you can select “Process Now” in the top right corner of the UI or you can wait for the automated internal job processing that occurs at 2:00am every night.

Monitor the Status field for the latest status of the processing. If the page doesn’t refresh on it’s own then you may need to refresh manually.

Once completed navigate back to Applications > Application Groups and search for the name of the Application Group Calculation rule that was created. In this case we searched for “JF” and will see all that match in a list view.



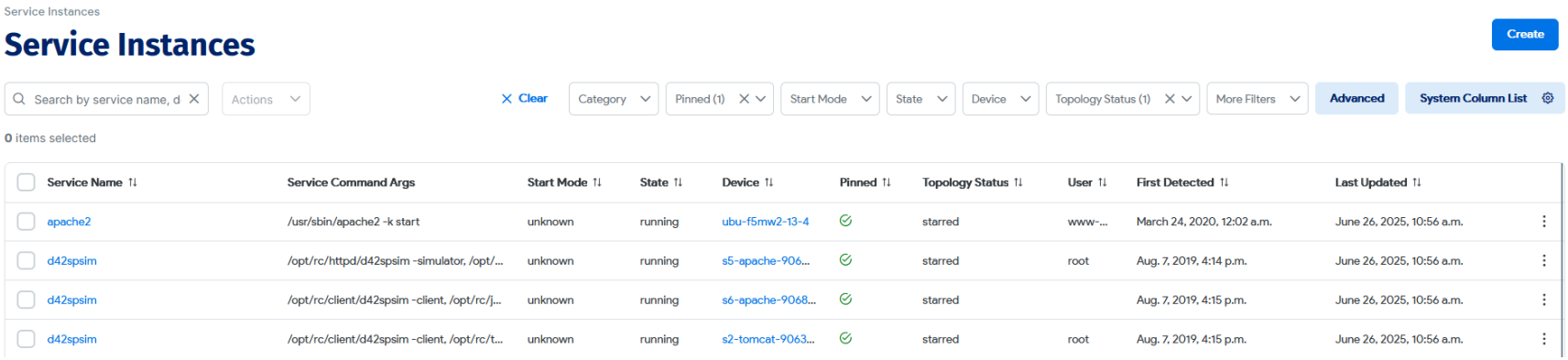
Click on the **Chart** hyperlink to the right of the Application Group that was created. The Devices selected within the Fixed criteria are isolated within the AppFocus “box”, this identifies the key systems the Application Calculation Rule was targeting.



My Application Groups

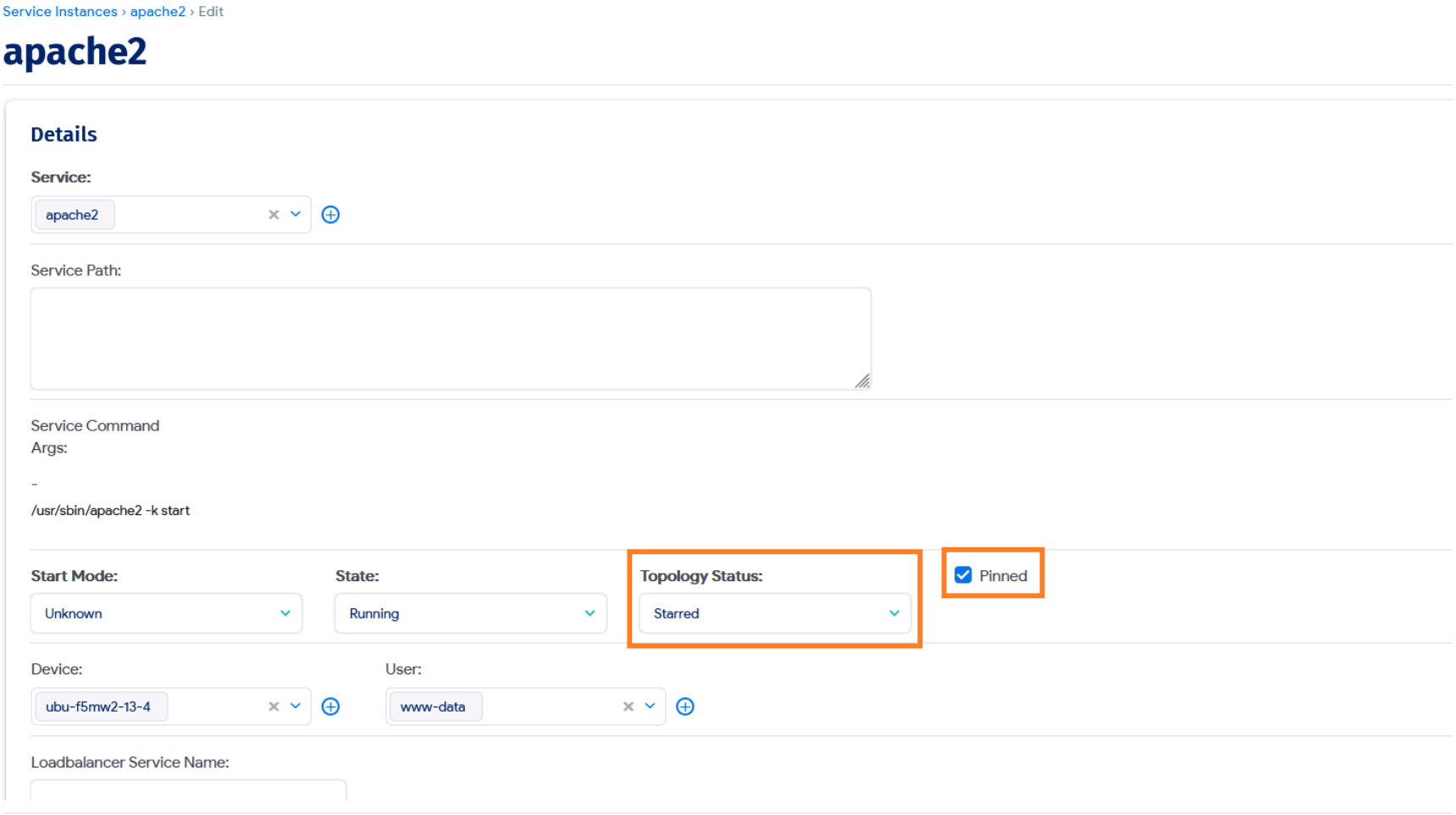
In this section we will look at the logic behind Application Groups and how the connections are established. Application Group starting points can be from a Starred service and can also target Resources, Application Components along with Service Instances directly. Typically you would start or end at the Database Server(service). However if your application does not utilize a database you will want to Star the starting service for your application. This can be done from the Topology view of the Device or from the Service Instances page by filtering on Topology.
If using DOQL queries for your Application Group Calculation Rule then you will need to verify your Service Instances are Pinned. Database, Application Layer and Web Server service instances are Pinned by default.

To check if a service is Pinned or Starred navigate to Resources > Service Instances. There is a column called “Pinned” where you can verify if Pinned or not. For a Starred Service Instance you must click on “More Filters” and select the “Topology Status” option where you can select “Starred” from the list. .

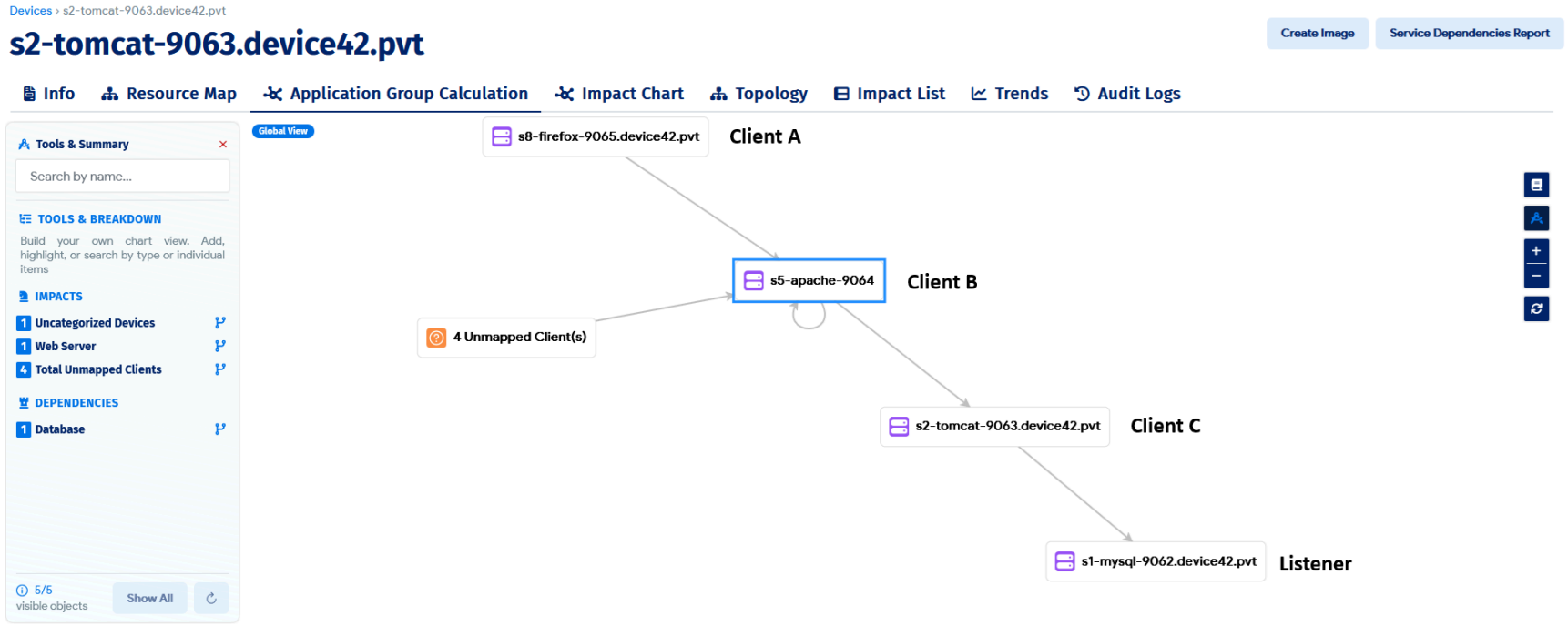


If you need to Pin or Star a service instance you can do that from the list view by selecting the service instance then from the Actions dropdown menu you can select to Pin the service or change it’s Topology Status.

Optionally you can modify the Service Instance directly from the Service Instance window itself.

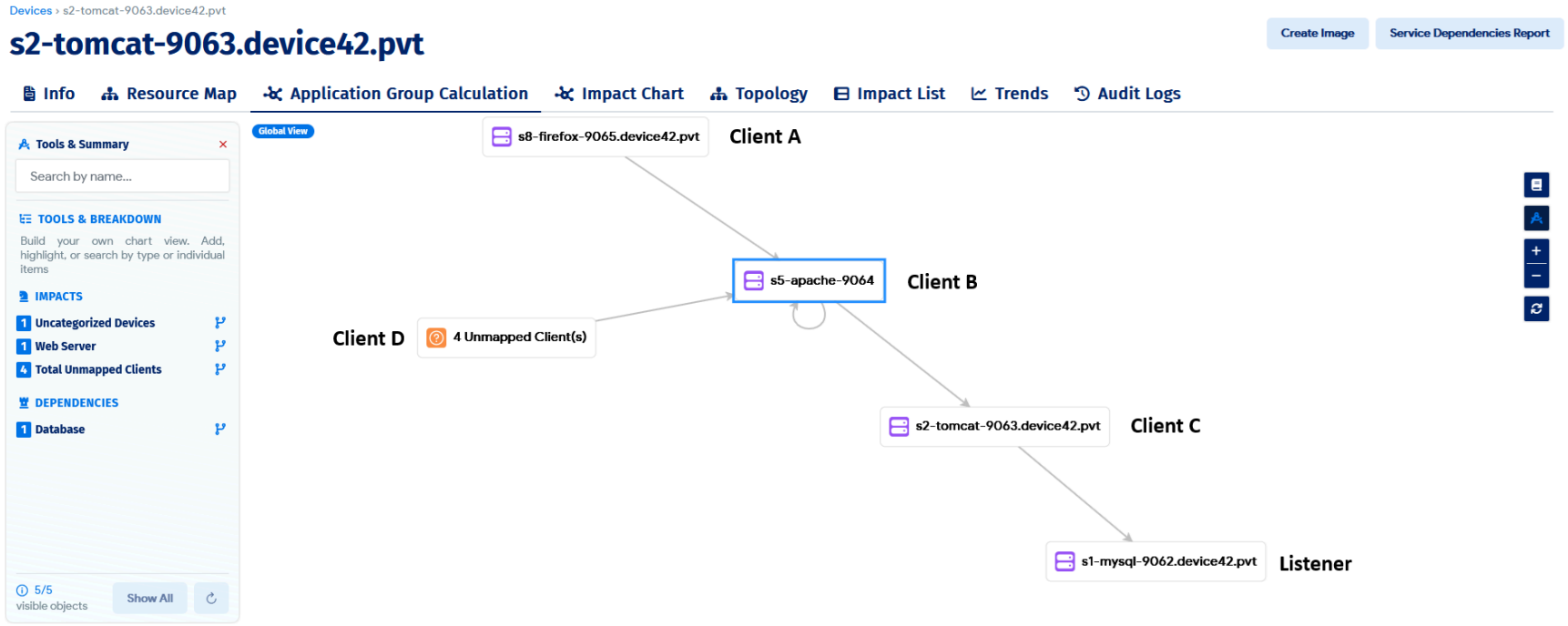


This is an example of how a basic Application Group is constructed. Depending on which type of Calculation Logic Template you are using in your Application Group Calculation rule the starting point will be with a pinned or starred service. Beginning with the pinned or starred service connections will be created by following the service connections. We can see the Service on Cient **B** has a client Service **A** connecting to it, That same Service **B** is also making a client connection to Client **C** which in turn is making a Client connection to the Listener at the end of the connection string. Device42 will keep following the Client Listen path of the services until we reach the end.



The addition of Application Component Templates to Application Groups allows forking of connections to take place. The above example follows through on a singular Service basis, making the connection strings linear. Because Application Components can be made up of multiple Services this allows connections of different services hosted on a server to be pulled into the Application Group.

This is the same example as on the previous page but this time you can see Client **B** has Client **A** and Client **D** connecting to it eventually making it through the chain to the Listener Device at the end.

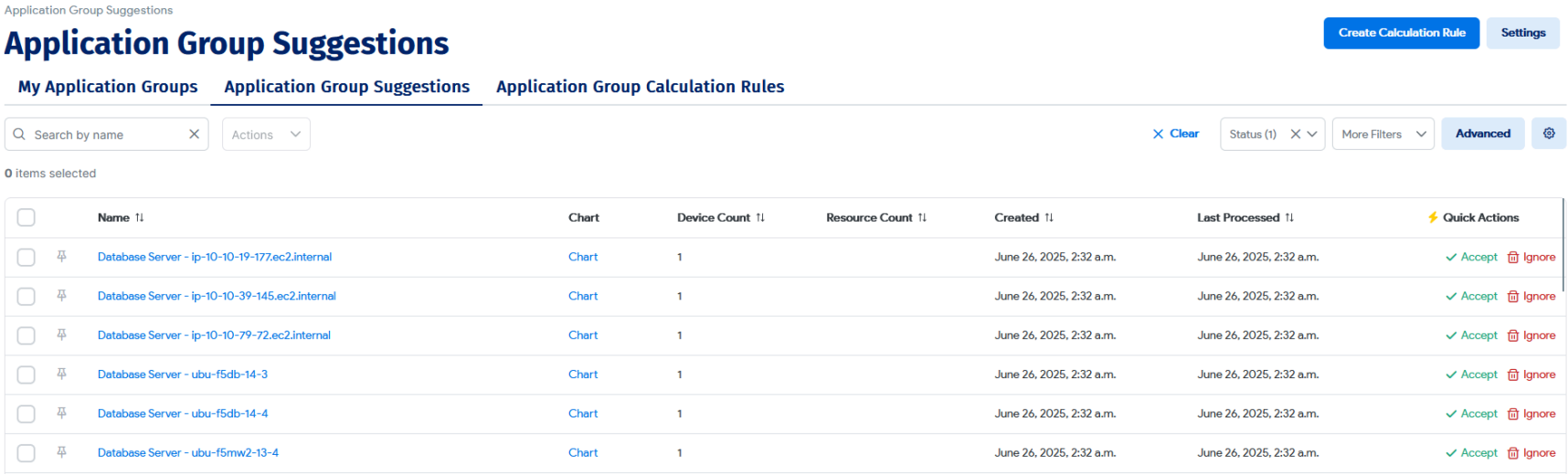


The Application Groups will also include multiple connections, If you have a Service in a connection chain that has multiple clients we will include all of these connections.

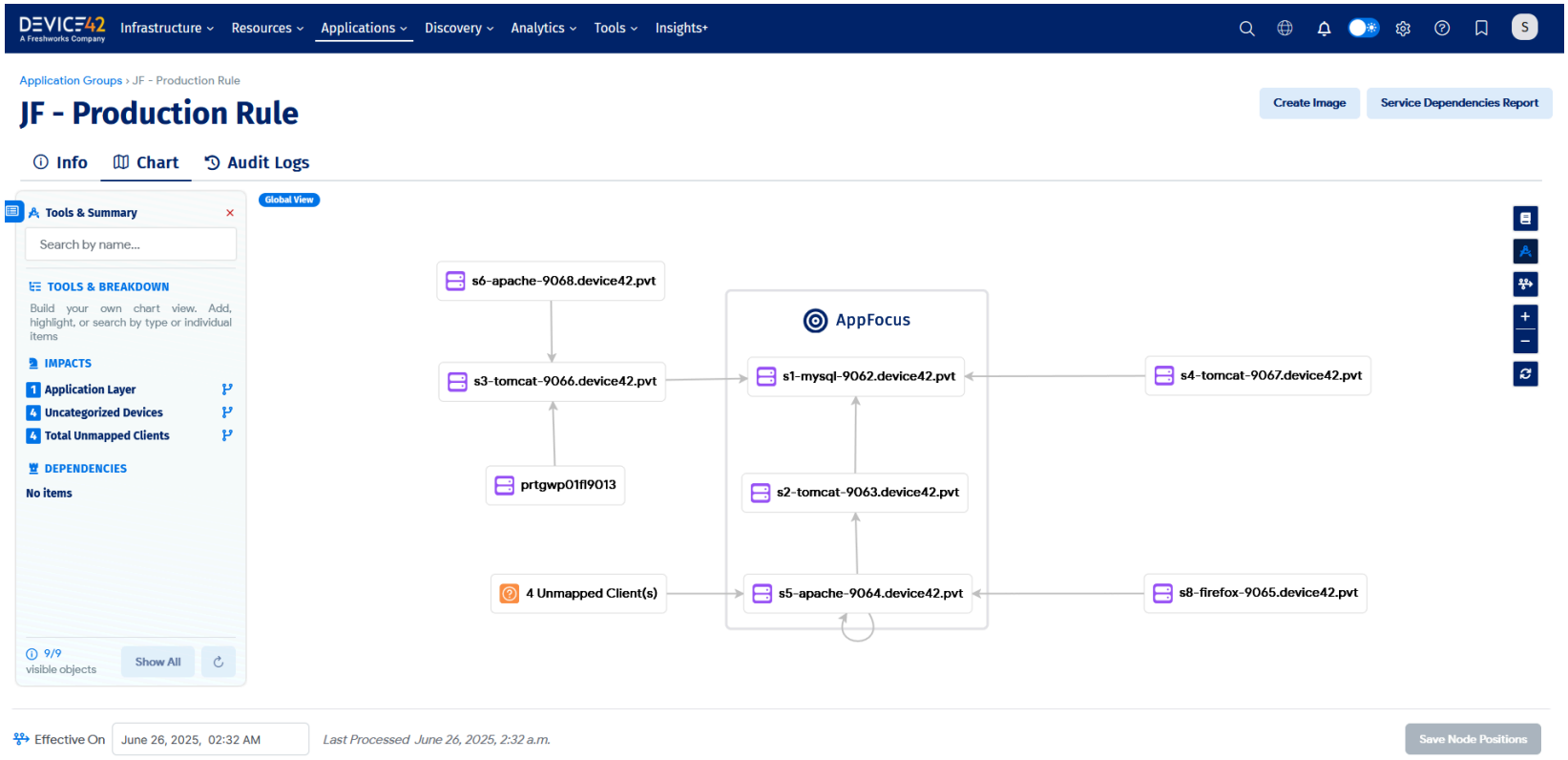
The Application Groups page has three tabs in it, My Application Groups, Application Group Suggestions and Application Group Calculation Rules.
The Application Group Suggestions page should be considered your working pallet, where you can review the Application Groups created by Device42. Once you have reviewed a Suggested Application Group and have decided you want to progress with it. You can Accept the Application Groups to move it from the Suggestion page into the My Application Groups page.

The My Application Groups page is where you store the Application Groups that you want to progress into a Business Service.

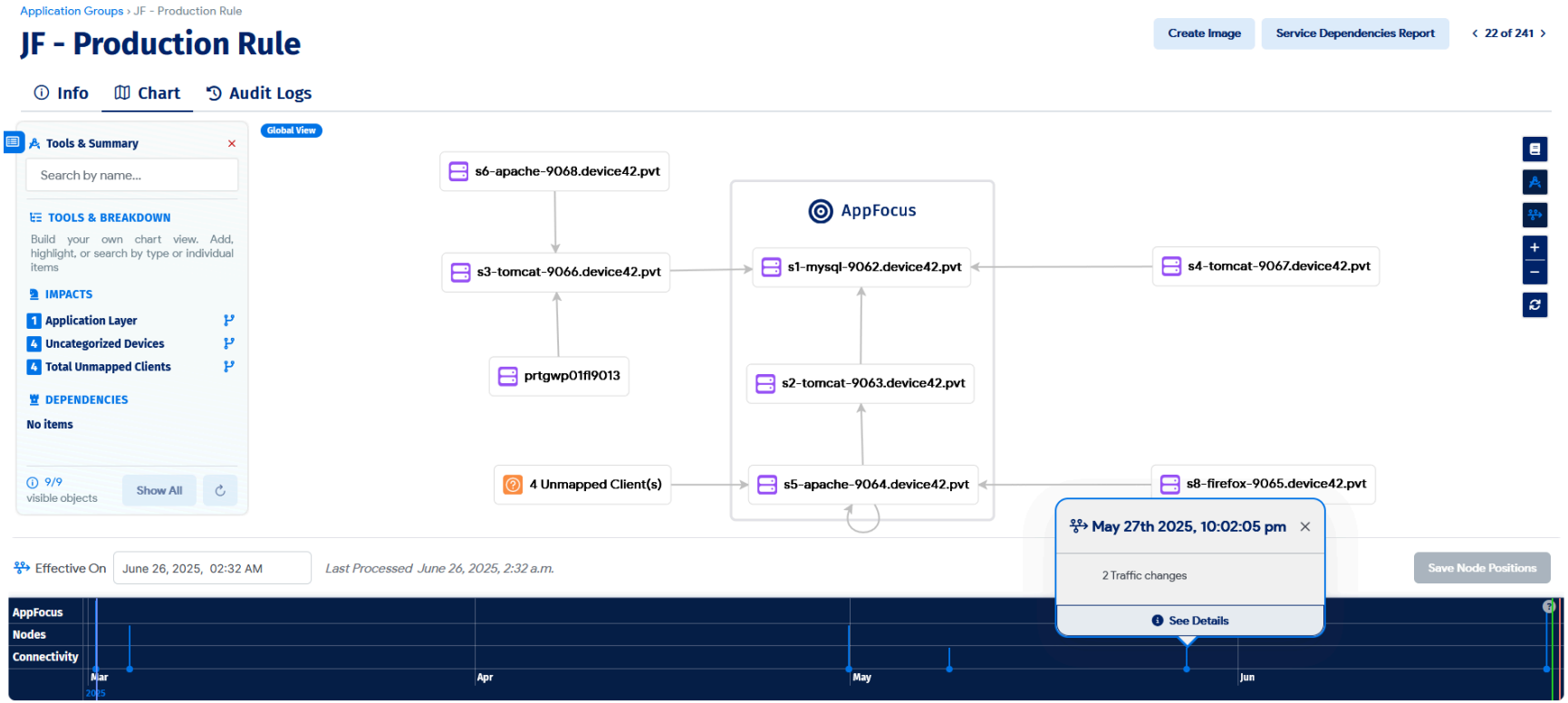
You can review Application Groups by clicking on the Chart button. This will load the Application Group for you to review.



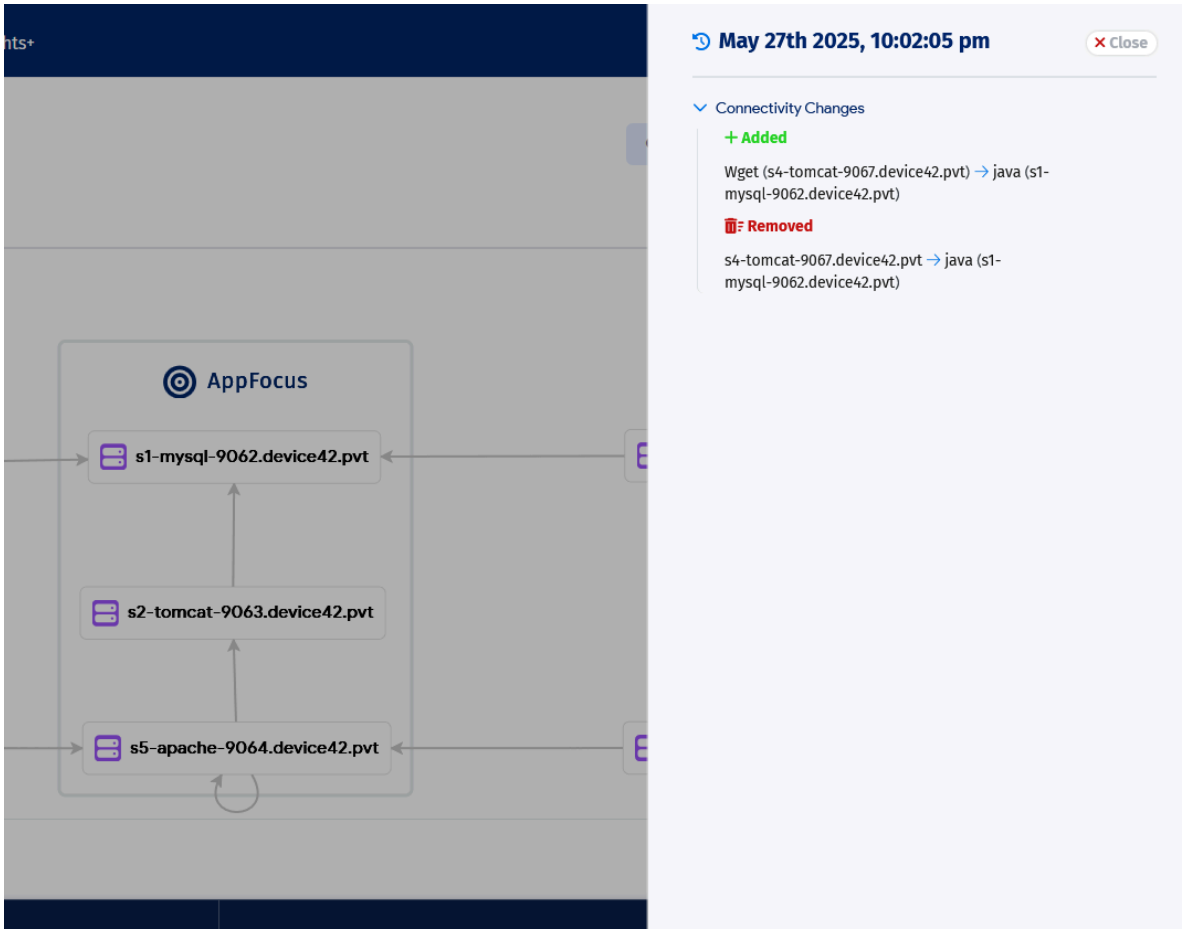
This is an example of what an Application Group will look like. The Devices found in the AppFocus box are ones that meet the criteria set in the Application Group Calculation Rule, In this case it is three systems identified as a single Business Service application in with Production set as the Service Level. Devices outside of this box have dependencies on these servers via a Service but do not meet the Application Group Calculation Rule criteria. These could be devices that do not have a Production service level, very useful for identifying Production applications that might be communicating with Dev or Test servers for example. They could also be Servers that have been added to the Application during its service life that have not been documented. You can investigate these devices to understand their relationship to the application and add the necessary options to include them in the AppFocus box.



Because the data captured for Application Groups is ongoing, Device42 allows you to see how an application in your environment has changed over time. This can be helpful for Incident Management and Change Control. There is a Time Settings button on the right hand side of the screen that when clicked will open a timeline at the bottom of the page, allowing you to scroll back in time and review where changes were made.

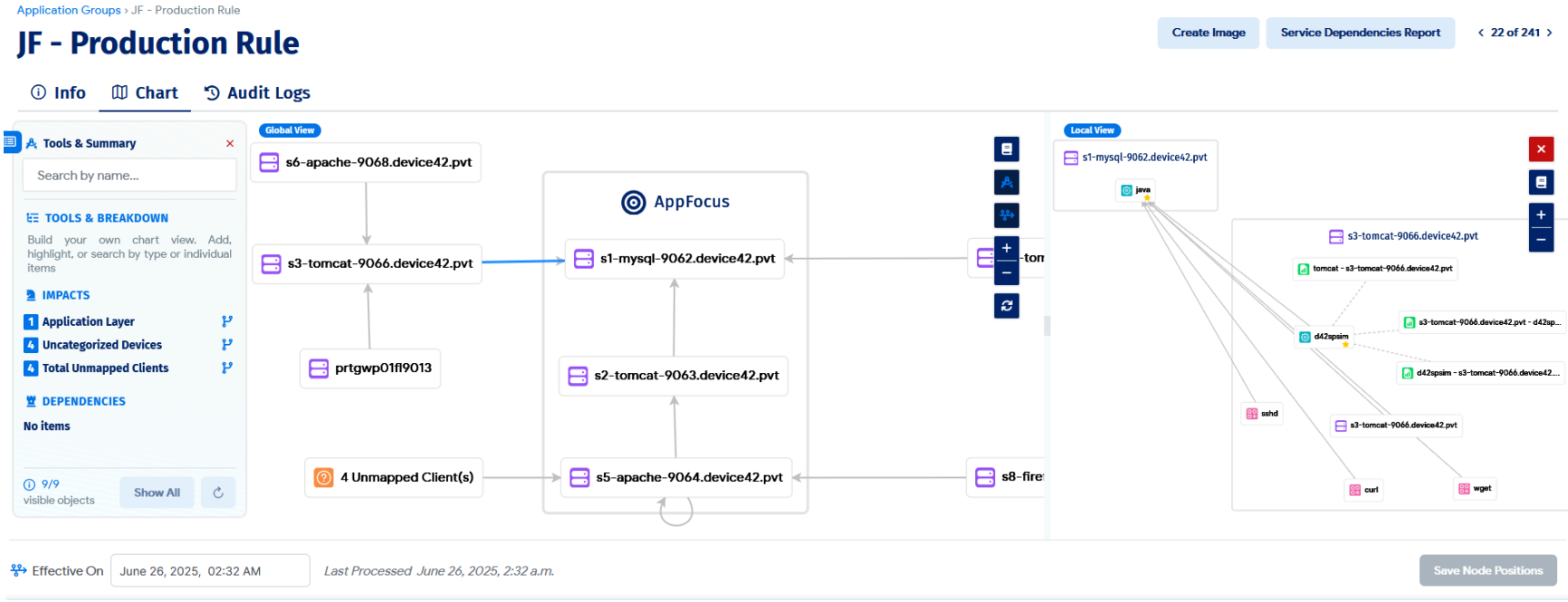


Clicking on See Details will show you the changes in more detail.

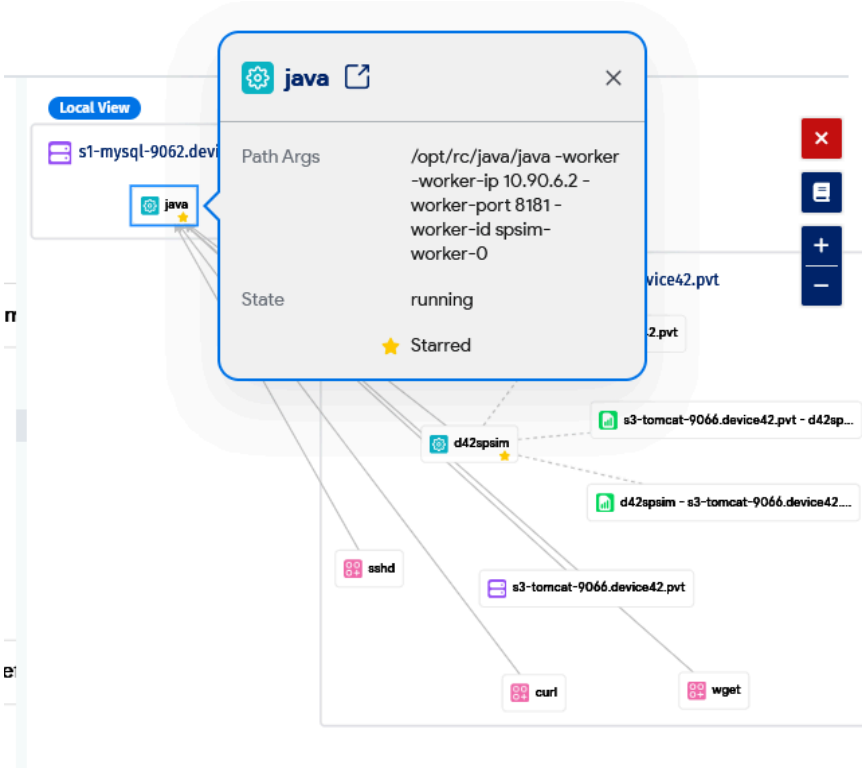


You can also drill into these views deeper to understand what service is creating this relationship in the Application Group. This can be done by clicking on any of the communication lines that connect the devices together.

We can see the services that make the relationship between the s3-tomcat-9066-device42.pvt system and s1-mysql-9062-device42.pvt system. The rose colored objects represent a Service and the blue objects represent an Application Component.



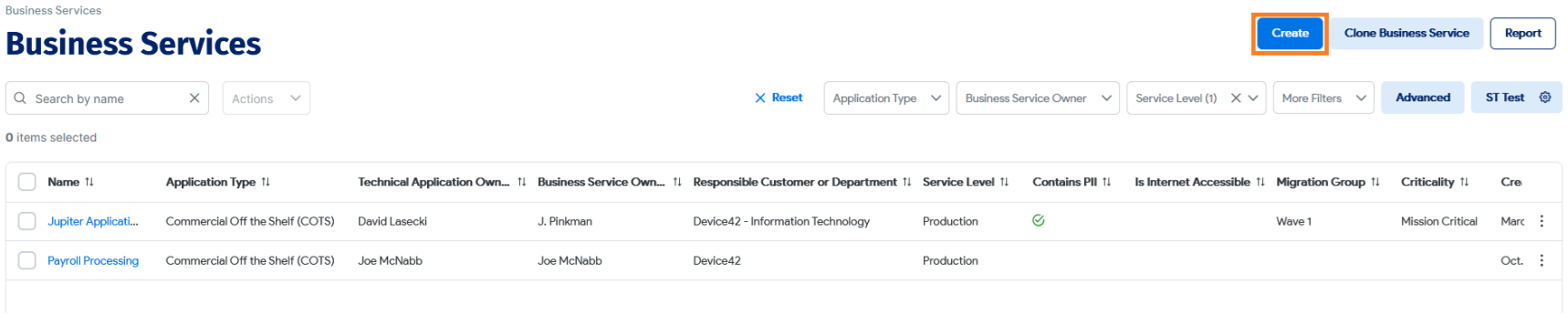
Clicking on the individual elements will open up a dialog box to get a snapshot of the information for that element. Clicking the pop out icon (square with arrow) will take you to that elements page to see it in more detail.



Business Services

Business Services are a way to solidify your understanding of an Application from an Application Group and create referenceable documentation. One thing to note is that Application Groups are dynamic and will update and change on new data that is discovered, Business Services are static and will not automatically change unless you have created Alerts for a Dependency removed or added. This is covered in more detail here: https://docs.device42.com/reports/reports/setup-alerts-and-notifications/#alerts-and-notifications

> To add your Application Group to a Business Service, first you will need to create the Business Service. This can be done from Applications > Business Services, Click Create to create a new Business Service.



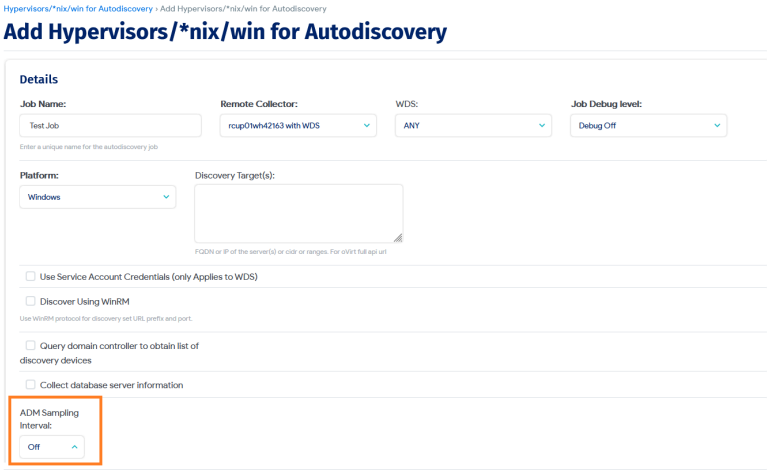
Recommended Steps for Success

1. Setup Discovery Jobs
2. Verify Data
3. Configure Application Groups
4. Configure Business Services

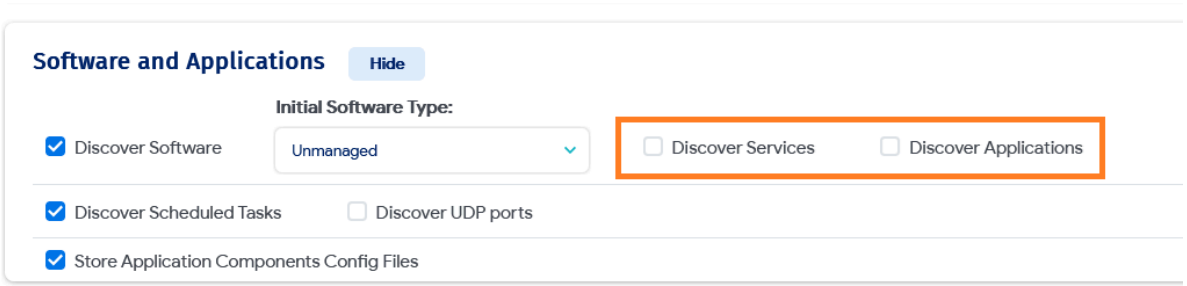
Setup Discovery Jobs

Begin by Discovering your Windows and/or Linux environment whether from a direct Windows or *nix Discovery job or from a TCP Port Scan job which will identify Windows and *nix systems on the selected network(s).
Before you schedule and Run the Discovery job be sure to update the ADM Sampling interval to **“Off”**.

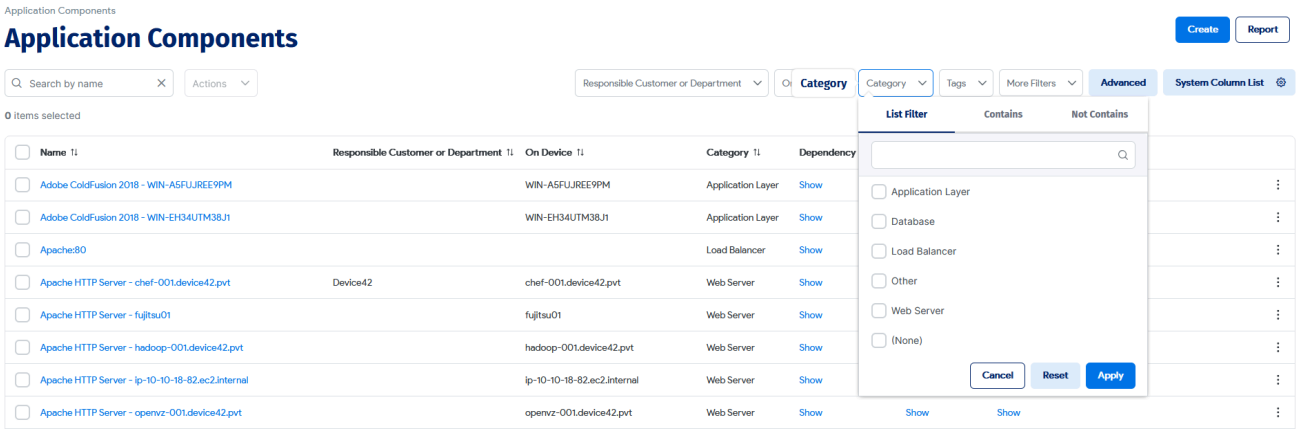
This will be enabled by default when licensed for Enterprise Application Discovery and Services Discovery.



Scroll down further in the Discovery Job settings and verify Discover Services is unchecked.



The reason for this is to first discover your environment and then identify the specific Application Servers you wish to perform the deeper level of Services and Application Component discovery on. By leaving the Discover Applications checked off, this will help determine which servers you may target for the AGs Discovery type jobs.
We recommend this approach because in most cases your ADMs licenses (Enhanced Application Discovery along with Services) are not equal to your Core licenses which can result in having your ADMs licenses allocated to servers that are not part of the Application inventory.
Once you've discovered your inventory, analyze the list of servers and identify the servers you wish to include in an Application Groups discovery job.
To help identify these servers you can navigate to Applications -> Application Components. You may have discovered some by Category of Application Layer, Database, and Web Server. You can also search by name.

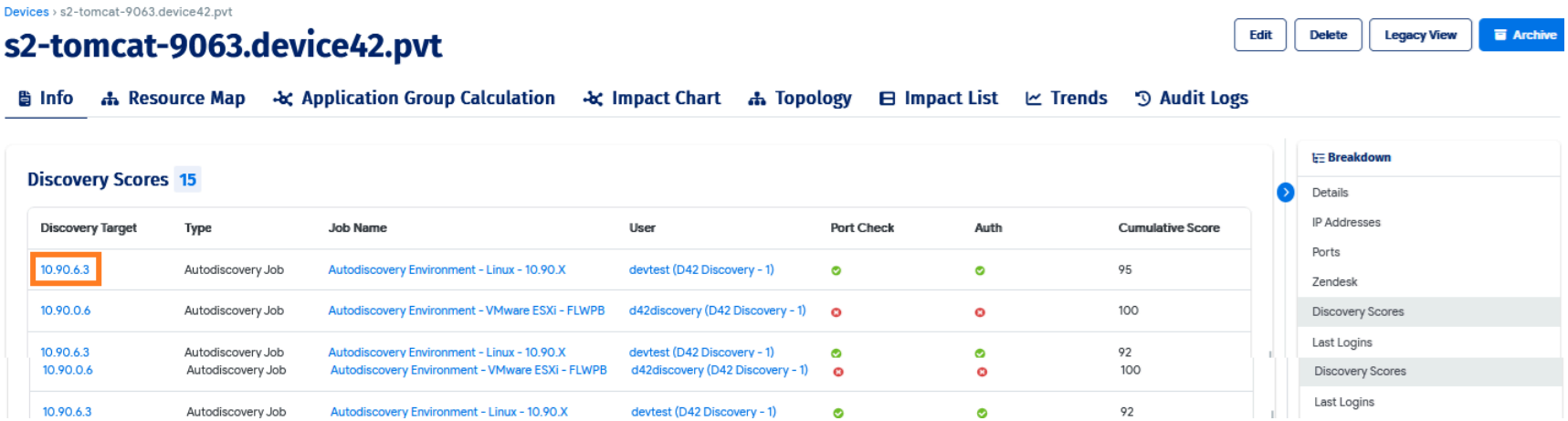


Once you've identified the Server(s) you will create a new *nix/Windows Discovery job for these systems with additional options selected to gather the communication information.

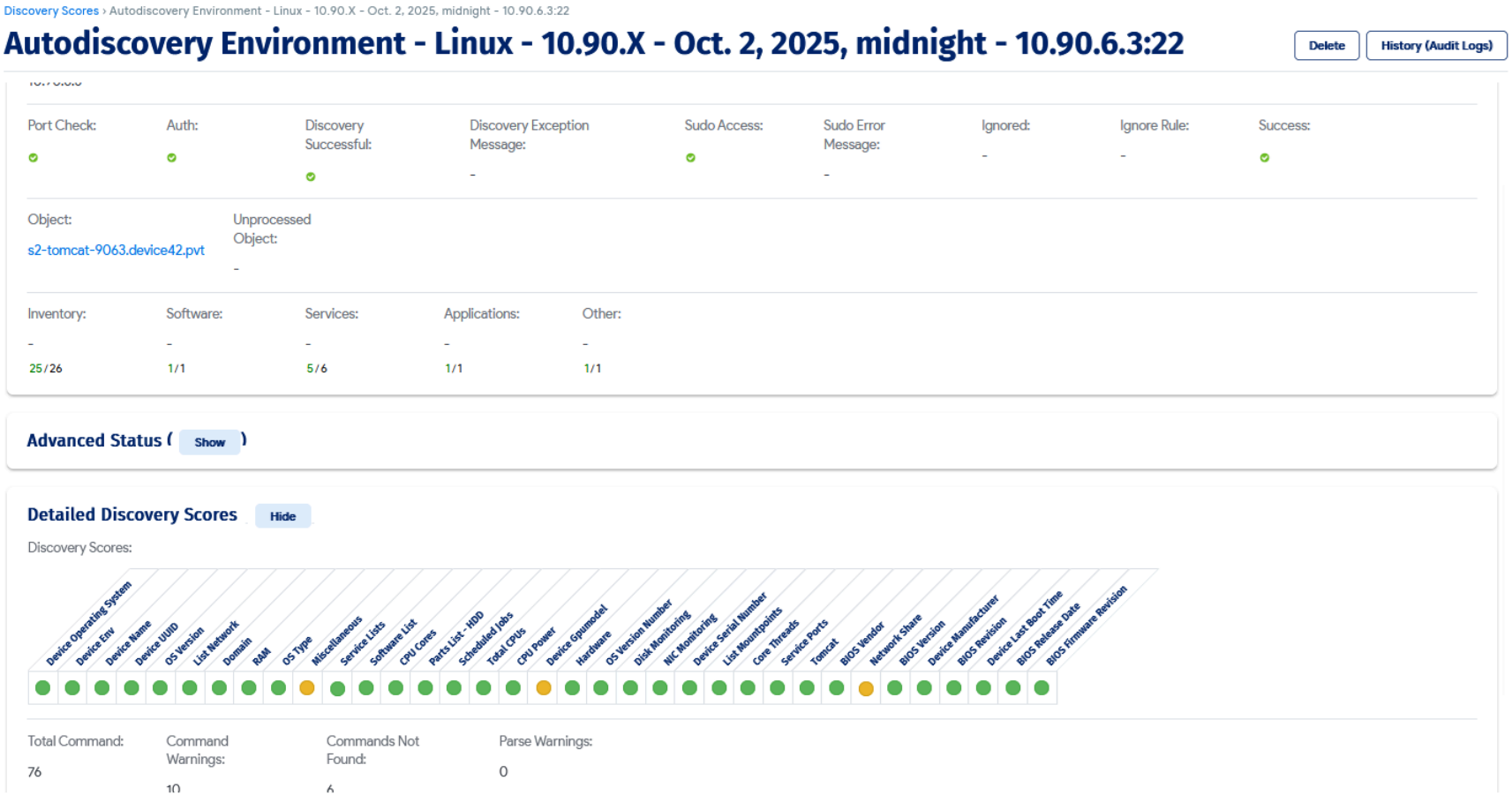
Create a new discovery job, select the correct platform, Windows or *nix and add the systems to the job.
Verify the ADM sampling is selected for 4 hours (recommended in the initial scans) along with the Discover Services and Applications checkbox is checked off. Schedule and Run this discovery job, when this job runs it will create additional jobs per server called Periodic Jobs.
These Periodic Jobs run at the 4 hour interval set and will perform the Netstat command to acquire the Listener and Client communications happening on the server(s).
It's recommended to wait about 24 hours to allow the Periodic Jobs enough time to gather communication information.

Verify Data

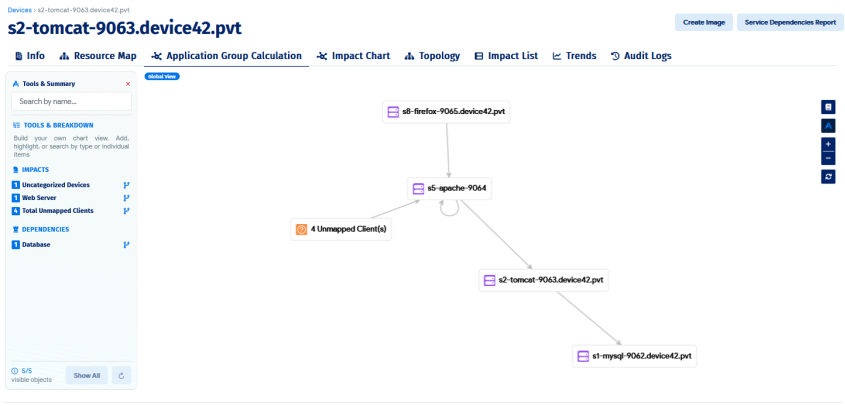
Next, verify the data discovered. Check the Discovery Score(s) of the target devices in the Discovery Job configured for ADM. This can be done by locating the Discovery Scores section of the Device record then clicking on the Discovery Target IP.

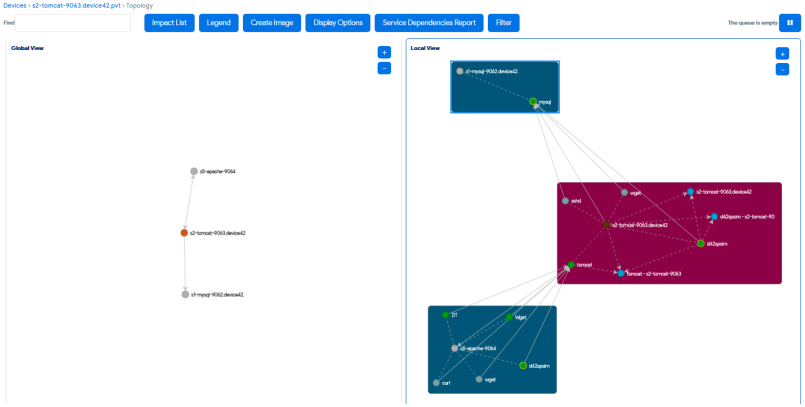


After clicking the Discovery Target, scroll down and click “Show” next to Detailed Discovery Scores, verify that Service Lists and Service Ports are at least Yellow if not Green. If the overall score is in the low 90’s or higher that is a good result.



Next, click on the hyperlinked Device under “Object” in the above screen. This will open the Device record. Check the Affinity Group Calculation and the Topology views. They should look similar to the screenshots below, Affinity Group Calculation is on the left, Topology on the right.





Configure Application Groups

The next step is to configure the Application Groups, start by creating an Application Calculation Rule, during creation of the rule select the Outcome, Suggest or Auto-create, next select the Starting Points **Criteria** (search with filters) or **Fixed** Devices, Resources, Application Components or Service Instances.

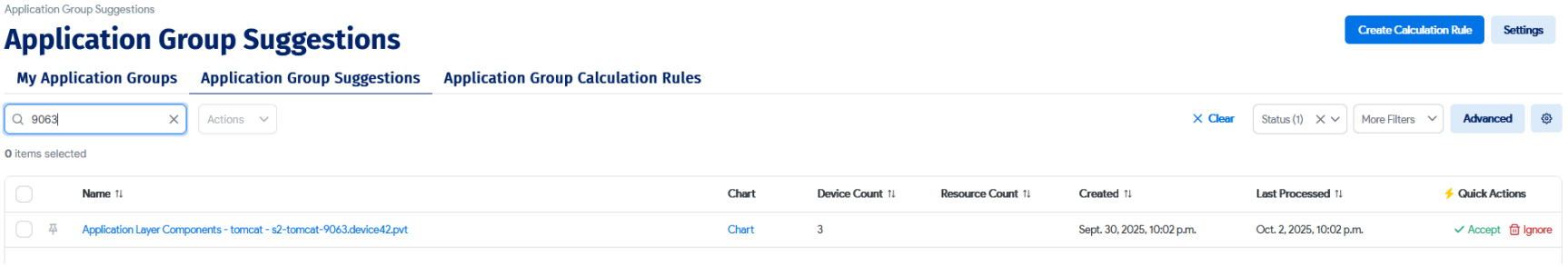
Next select how the Devices should be Grouped by Name, Service Level or some other criteria.

Select the Calculation Logic Template, this will be a Form based or DOQL based logic.

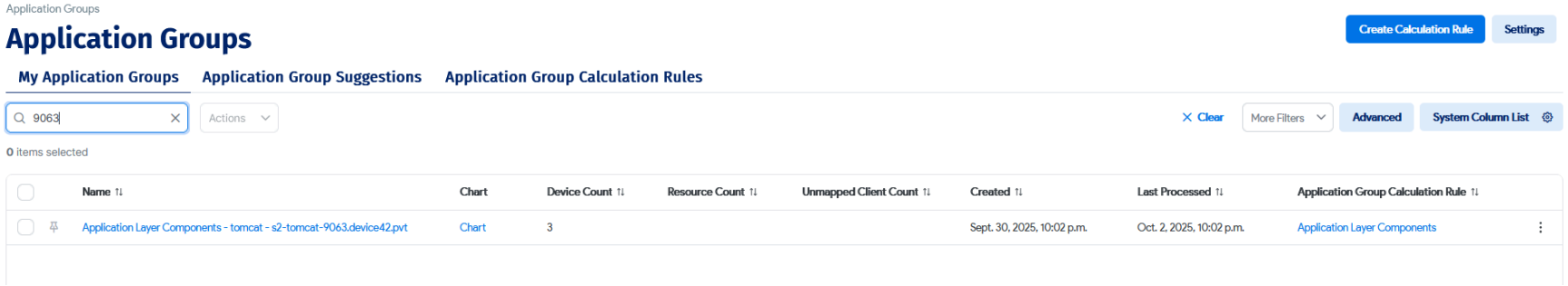
Save the configuration and you can select Process Now.

NOTE: If you have just began discovering these Devices and their communications you may not get positive results right away. You may need to wait several days for Communication information to be captured so it can be calculated properly.

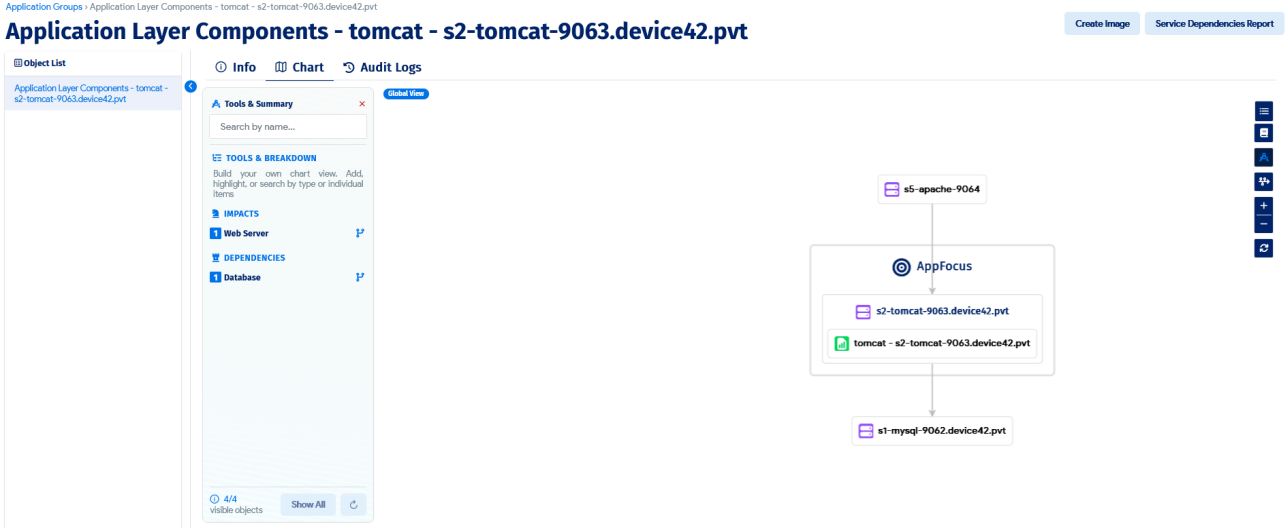
Once the processing has completed and assuming there has been sufficient communication data collected you should see your Application Group Suggestion created where you can Accept or Ignore it,



Accepting it will move it to “My Application Groups” where you can create a Business Service with the calculation that was generated.

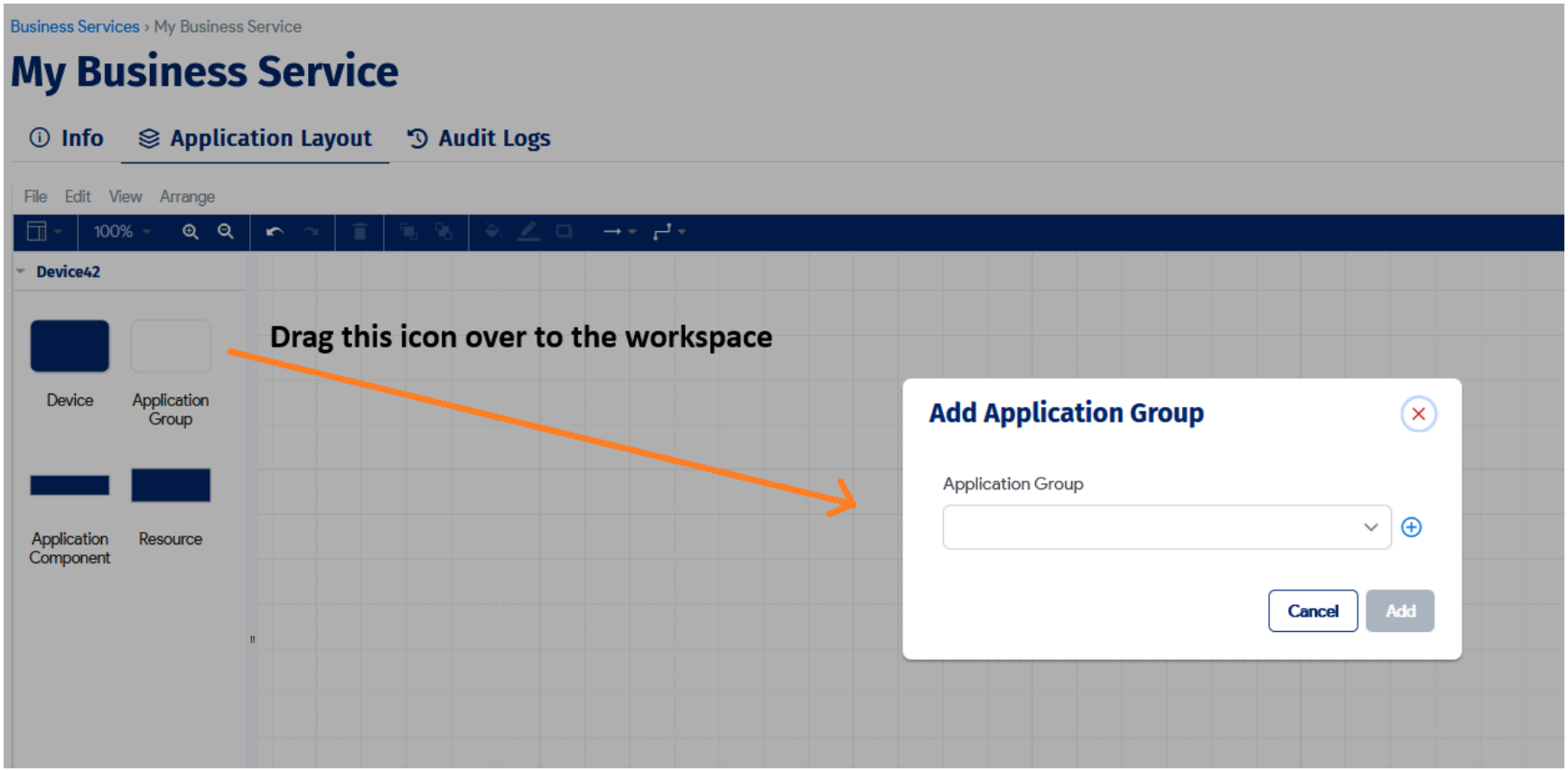


Clicking the Chart link will bring you to the image below.

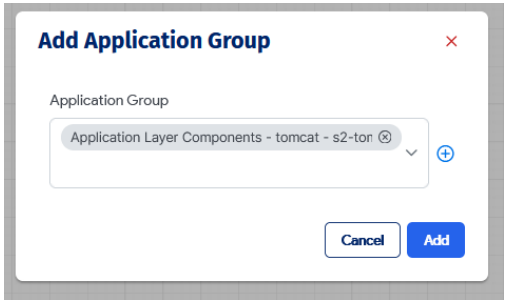


Configure Business Services

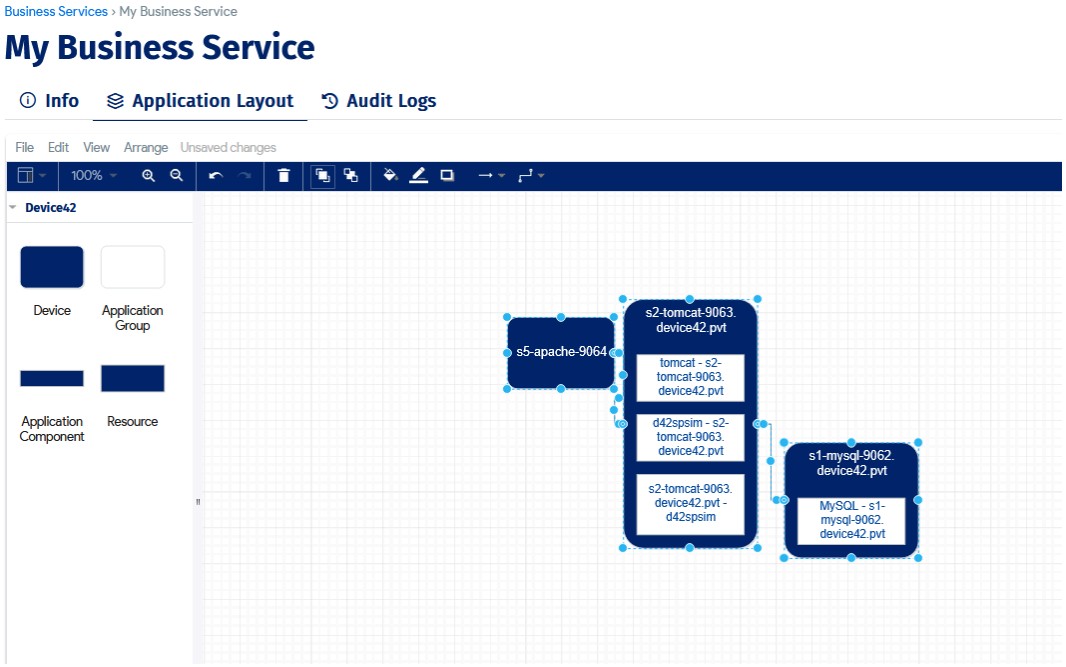
Business Services can be created directly from the menu or from an Application Group. From the menu simply navigate to the menu item, open Business Services and click the “Create” button on the upper right corner. Fill out the necessary fields, at least the name and save. Once saved click the Application Layout tab then click the “Edit” button in the upper right corner to begin adding objects to your Business Service, you can add Devices, Application Groups, Application Components and Resources. Next, add an Application Group to this Business Service by dragging the Application Group icon to the workspace.



You will be prompted to search for an Application Group, begin typing a name and once it's been found click on it to select it.



After you click Add you will be presented with a screen like this.



At this point you can modify the layout within the Business Service, add colors and shading to objects to make the stand out. For more information refer to the Documentation.

Troubleshooting

Here are some key areas to look for when troubleshooting Application Groups.

Verify Discovery Settings:

Ensure that the discovery jobs are configured correctly, with the correct credentials, IP ranges, and options selected.
Check that the ADM Sampling Interval is set appropriately for your environment and use case.
Check Network Connectivity:

Verify that there is network connectivity between the Device42 main appliance/remote collector and the target devices.
Ensure that firewalls or network security settings are not blocking the discovery traffic.

Verify AGs License Usage:

Visit Tools > Licensing and ensure that the ADM license discovered count is below the licensed amount.
If you find you've exhausted your ADM licensing on the wrong systems there is a Webinar that shows how to reclaim them.
Introduction to Ensuring Success with ADM License Usage and Management

Review Discovery Logs:

Check the discovery logs in Device42 for any errors or warnings that might indicate issues with the discovery process.
Look for messages related to authentication failures, timeouts, or other connectivity issues. Go into each Discovery Job and click 'Show' in the job results breakdown. Turn on Payload Diagnostics > Tools > Global Settings > Payload Diagnostics then review them in the Payload Diagnostics section of the Discovery menu.

Validate Service Detection:

Confirm that services are being detected correctly on the target devices. Inaccurate service detection can lead to incomplete or incorrect Application Groups.

Use the "All Services" and "Service Instances" pages in Device42 to review the detected services and their instances.

Ensure Key Services are Starred when using the Form Logic or Pinned if using DOQL Query based Logic:

Verify that key services for applications that communicate with other devices/services are starred or pinned in the Service Instances section. This ensures that they are used as starting points for Application Groups.

Check Application Component Creation:

Visit Applications > Application Components to ensure that application components are created based on the application component template rules for finding services.

Review the Application Components Template section to confirm that the rules to create Application Components are valid.

Examine Application Groups:

Verify that Application Groups are being generated correctly and represent the expected application dependencies. Applications > Application Groups. Check both the auto-created groups and the suggested groups.

Change the Application Calculation Rule or DOQL that is used to create Application Groups as needed. If the Application Groups do not show any or enough connections you can modify the “Limit of connections” in the Calculation Logic Template from the default setting of 20 to something higher such as 60 or 80. If a DOQL query is being used, ensure client OS device communications are not being calculated so we only see Server OS communications, or lower the date range for discovered services. Customizing the DOQL(s) can be used as necessary for more specific fine-tuning.

Use the Application Group Calculation Rules to refine the focus of the Application Groups if necessary. If you expect more than one device to be inside of an AppFocus box, then change the Group By in the Application Calculation Group rule to “Service Level” instead of “Name”. This Group By clause will create multiple Application Groups based on the rule. If we are grouping by name, this will include devices with unique names and each device with a unique name will have its own Application Group, whereas when you choose group by Service Level this will make one affinity group for all ‘production’ service level devices, and allows multiple devices to be included in the same AppFocus box. Custom Fields for devices can be used in the “Group By” logic as well.

Check Database Discovery:

If database discovery is part of your Application Group process, ensure that database instances and their connections are being discovered accurately.
Review the "On-Prem Databases" page to check the database count and connection count.

Check UCS/ACI/Load Balancer Discovery:

Ensure the Load Balancer discovery jobs are set up properly and executed without error. Ensure the order is SNMP jobs first then UCS/ACI/Load Balancer discovery scheduled after the SNMP discovery job(s) when targeting Load Balancers.

Update Device42:

Ensure that your Device42 instance is running the latest version, as updates may contain fixes and improvements for the AGs discovery process. Visit Tools > Update, this will show your current version as well as provide a link to check the latest version available.

Consult Device42 Support or your Customer Success Team:

If you encounter persistent issues or complex problems, reach out to Device42 support/CS for assistance. They can provide expert guidance and help resolve specific challenges.