Presto 2.5

Using Presto Rules

Table of Contents

So, I Installed Presto. Now What?

Don't be intimidated. Writing Presto rules is simple, once you get the hang of it. Let's start by looking at the default rules that are in place when you run the Presto Server for the first time. Open the Presto Server Console and navigate to the **Rules** pane (click in the world **Rules** in the leftmost column). You should see this (Figure 1):
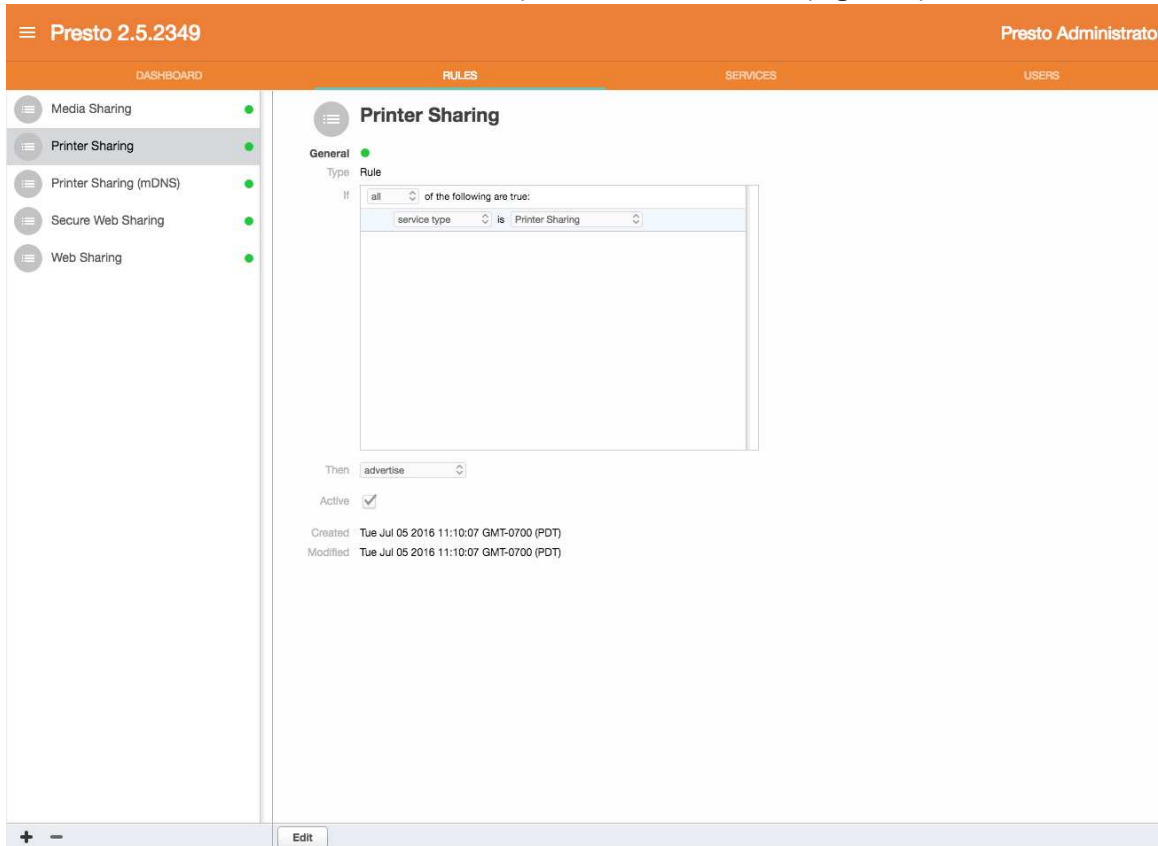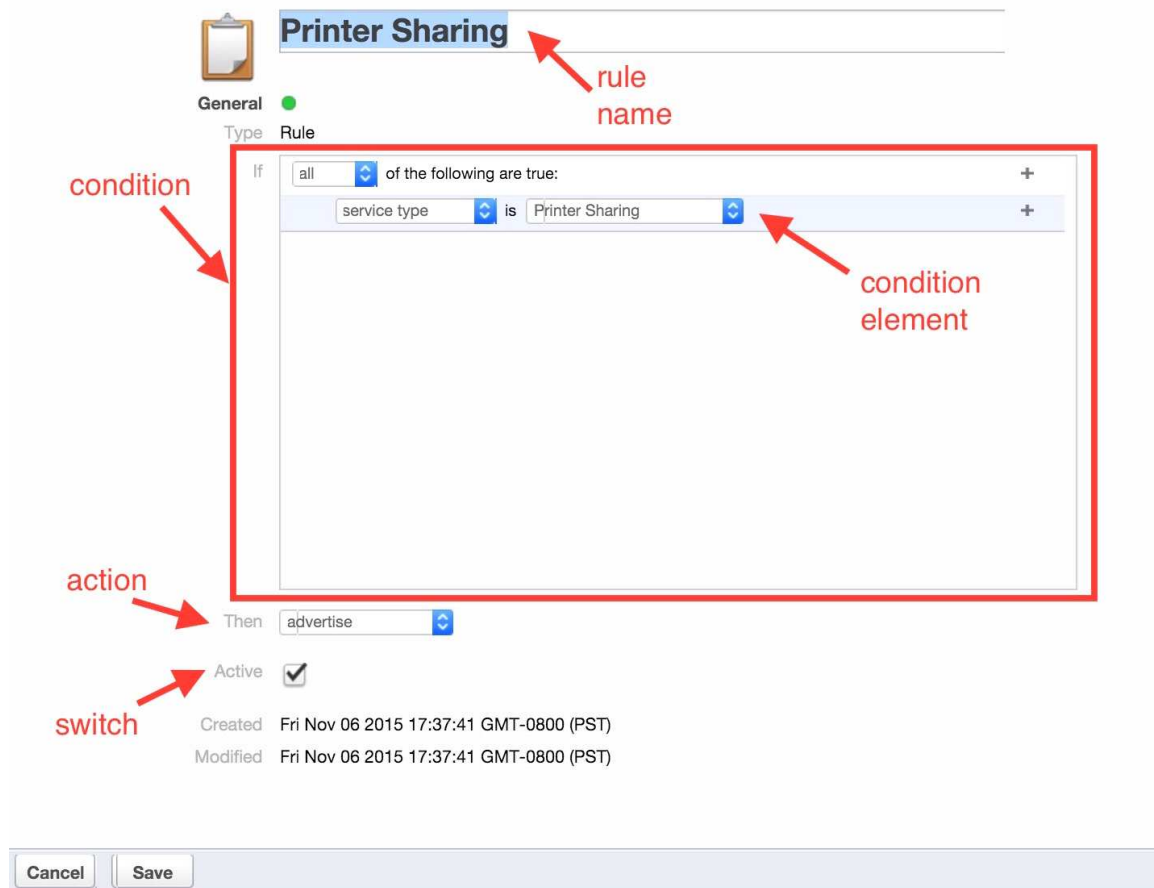

Figure 1: Presto Server Console with Default Rules

Default Rules

Select the **Printer Sharing** rule by clicking on it in the second column. Click the **Edit** button in the lower left corner of the **Rules** tab. You should see something that looks like Figure 2 (minus the red annotations).

Figure 2: Parts of a Rule

Rule Name

The rule name is simply that. When the rule is open for editing, you can change the rule name. Rule names must be unique, but the rule editor will force uniqueness by adding a number if you try to make two rules with the same name.

Condition

In Figure 2, everything enclosed in the red box is the rule condition. Note the word **If** in the upper left corner of the box. A condition is made up of condition elements.

Action

Click on the rule action next to the word **Then** directly below the rule condition box. Clicking on the arrows opens a pop-up menu of two choices: **advertise** and **do not advertise**. We'll talk more about this in a moment.

Switch

If the switch **Active** check-box beneath **Then** is checked, the rule is active—it will have an effect on how Presto responds to queries for services. If it isn't checked, the rule is inactive, and will have no effect.

What Does This Rule Do?

In order to explain what this rule does, you must first understand what Presto is doing in a larger sense. One of Presto's primary jobs is to respond to queries for services. For example, to print a document from your mobile device, you open the print dialog and can select a printer from a list of available printers. For your mobile device to find available services like printers or AirPlay devices to list, it performs a browse operation. With Presto installed on your network and device, the browse operation takes the form of a query message sent by your device to the Presto server. This query message specifies the type of service your device is browsing for (for example, printers), and may include additional context, such as the user ID of the person performing the browse or the location of the device sending the query message. Presto server receives the query and applies the logic expressed in the active rules to determine which of the available services on the network will be included in the list you see on your device, or *advertised*, in response to the query. (When a service is included in a query response, that is called *advertising* the service).

Rules match services to queries on a pairwise basis—one query and one service are considered together when evaluating the rule condition. Each condition element in a rule is a statement about the query or service (or sometimes both) that is either true or false. If all of the condition elements in a rule are true when evaluated with a particular query/service pair, that query/service pair is said to *satisfy* the rule. The effect of the rule on the response to the query depends on the rule's action. For the moment, let's focus on rules with the *advertise* action. If a rule that advertises is satisfied by a particular query/service pair—that is, the condition of the rule, taken as a whole, evaluates to true for that query/service pair—then the service will be included in the response to the query, or advertised. In this example, the printer (the service) will be one of the choices presented in the print dialog box when you browse for printers on your mobile device.

Returning to the question—what does this Printer Sharing rule do? Printer Sharing, and the other default rules (except Printer Sharing mDNS; ignore that one for now) are the simplest rules that can be written for Presto. They all have only one condition element—the service type. The service type condition element applies to both the query and the service. Every query includes the type of service that is being browsed for. Likewise, every service has a type. When the Presto server receives a query of type *Printer Sharing*, the rule engine pairs this query with every printer currently available on the network, and evaluates this rule against the pair. Because the rule doesn't make any other statements about the printer under consideration, all printers will satisfy the

rule when paired with any query of type *Printer Sharing*. The Printer Sharing rule will result in all available printers being advertised in response to any browse for print services. Each of the other default rules except Printer Sharing mDNS is similarly promiscuous—all services of the type specified in the rule will be advertised in response to any query for that type. For smaller organizations with only a few printers all in the same location, this may be the desired behavior.

Maxims

Before considering more complex scenarios, there are some important maxims that must be observed. The first condition element in any rule is a compound condition element—it groups together a set of condition elements beneath it. Compound condition elements come in two flavors—*any* or *all*. We'll discuss how these are used in more detail later. For now, just accept the first Maxim:

> Maxim 1: The first condition element in any rule must be a compound condition element of type *all*.

You can try to change it to *any* if you want, but when you save the rule the editor will respond with an error message and refuse the change.

> Maxim 2: The second condition element in any rule must specify the service type that the rule applies to.

The editor will enforce this maxim, displaying an error message and refusing the change when an attempt to save is made.

The rest of this document will focus primarily on different scenarios that illustrate how rules can be used to achieve desired effects. In each case, emphasis will be placed stating the desired behavior as clearly and precisely as possible. Our recommended best practice is to do the same. Let's just make that a Maxim:

> Maxim 3: Before you start composing rules, take the time to define what you want them to do, clearly and precisely.

Scenario 1: Different Printers for Different Groups of Users

Operating systems typically require users to be assigned to groups, allowing administrators to grant or deny permission to resources based on group membership. In this scenario, the term *group* is used to refer to the group identities managed by the authentication and authorization mechanisms of the underlying operating system.

In this scenario, we make the following assumptions and statements of intent:

Our organization has an Art Department. Employees in the Art Department belong to

the group *art_dept*.

The Art Department owns printers used for graphics production. These printers use expensive media and consumables. Only employees of the Art Department should be able to print on these printers.

All other printers should be available for use by anyone.

Stated in terms of sets:

- We have a set of people called *employees*.

- We have a subset of employees, called *art department employees*.

- We have a set of *printers*.

- The set printers is divided into two disjoint subsets –

    - *public printers*

  AND

    - *restricted printers*, where restricted printers are those belonging to the Art Department.

We need a way to identify sets of users and sets of services (for example, printers), and a mechanism for writing rules that control advertisements based on membership in those sets. Fortunately, Presto provides just such a mechanism.

To illustrate a point, first we'll write a set of rules that, while functional, are overly complicated. Then, we'll see how rules can be simplified in some cases.

Tags

In the Presto world, tags are your best friend. Tags are simple character strings that can be associated with users, queries, and services. Some tags are automatically generated and assigned by Presto. Administrators can define their own tags and assign them to users and services. Rules can include condition elements that depend on certain tag values being associated with queries or services.

Group Tags

Presto automatically generates tags based on the groups in which a user has membership. Figure 3 shows a Presto user profile with automatically generated group tags. Each tag name is created by appending the group name to the string *group:*.
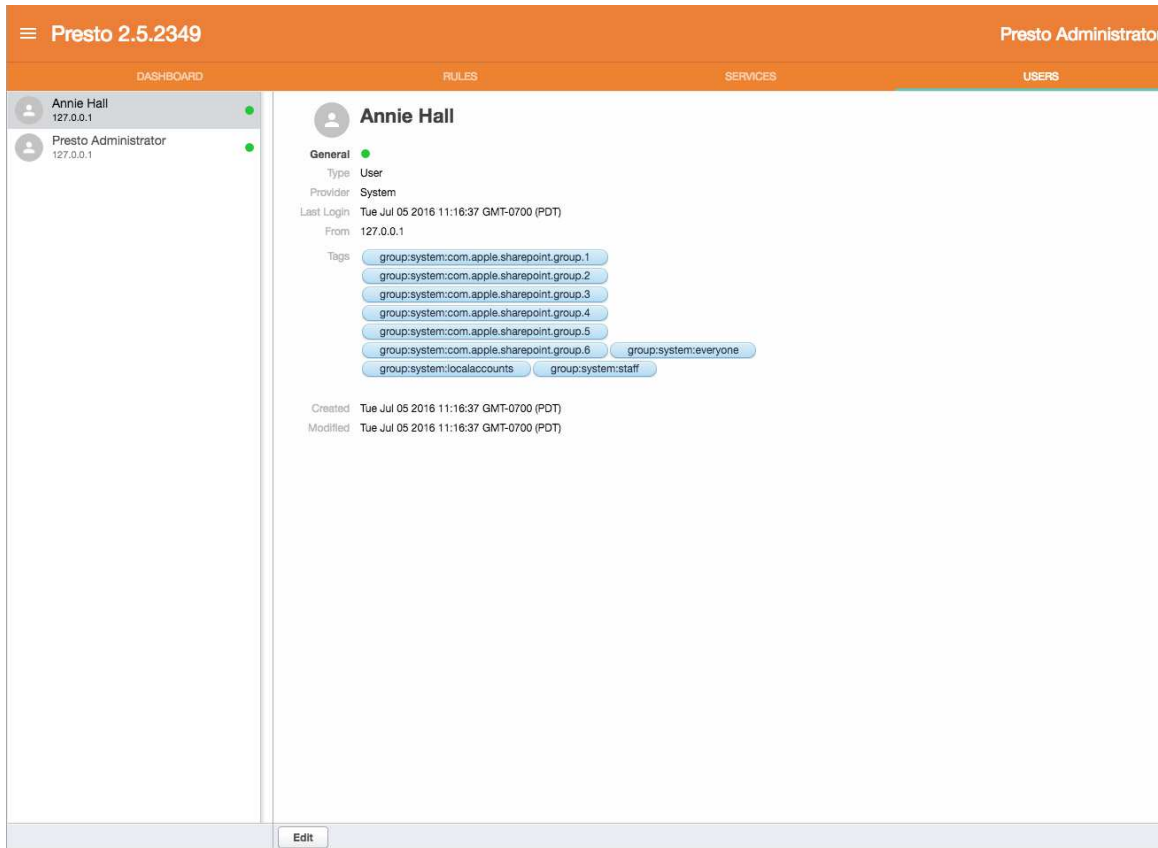
Figure 3: Automatically Generated Group Tags

If a user is logged in to the Presto app on their mobile device, and they are using a device that supports user identity in queries, then Presto will include all of the user's tags in the query context.

Now we can assign tags to printers to distinguish between public and restricted printers. Tags are assigned through the **Services** pane, as shown in Figure 4.
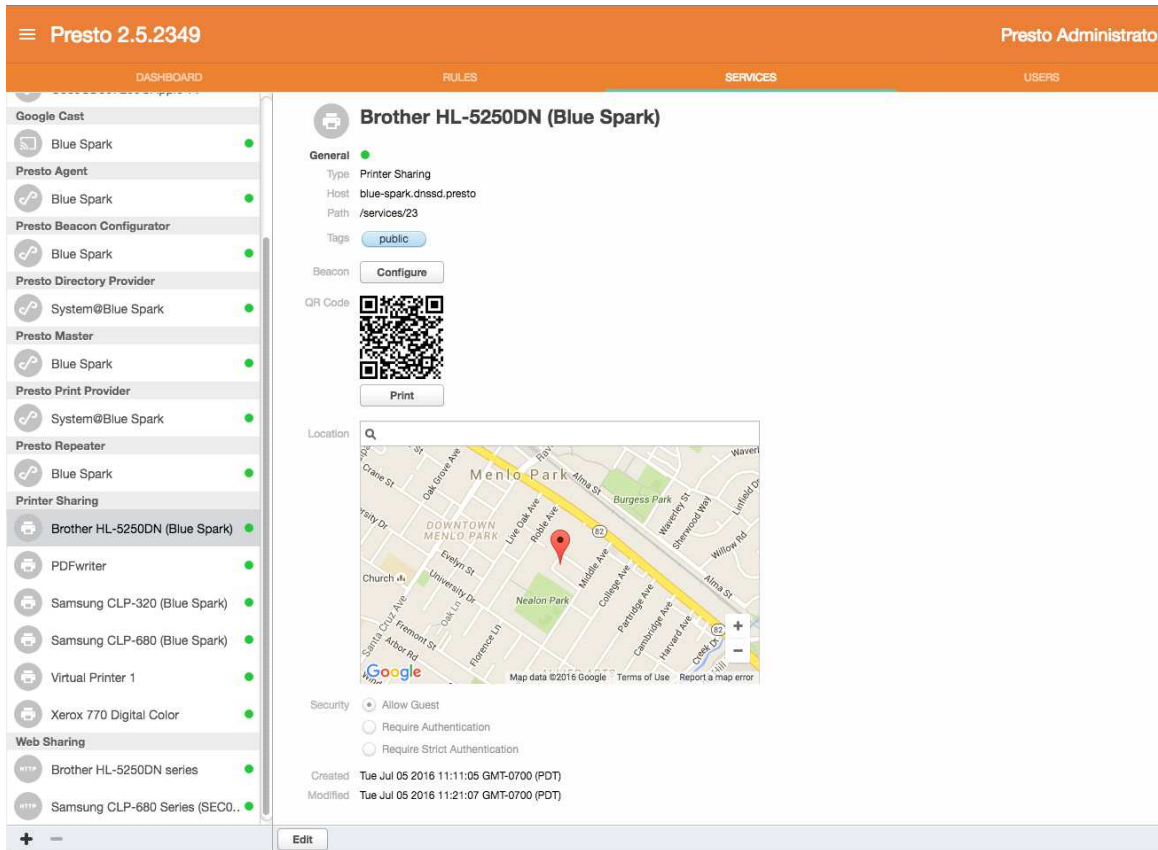
Figure 4: Assigning Tags to Services

Presto offers auto-complete options when entering a tag name, if the entry matches existing tags. Let's assume that the other printer (the Samsung) belongs to the Art Department, so we assign it the *restricted* tag (see Figure 5).

Now we can create rules that provide the desired behavior. The rule Public Printers (in Figure 6) will advertise printers with the *public* tag to all queries for printers. Note that we don't specify any user tags because this rule will work for any query,

Figure 5: Another Service Tag

regardless of which user originated it. The rule Restricted Printers (in Figure 7) will advertise printers tagged with *restricted* only to queries from users with the tag *group:art_dept*. So, for example, if Annie in the Art Department tries to print a document, all printers will appear in her list of printer choices. Another user who is not in the Art Department (or who is not logged in to the Presto app on their device) will only see the printers tagged *public*.
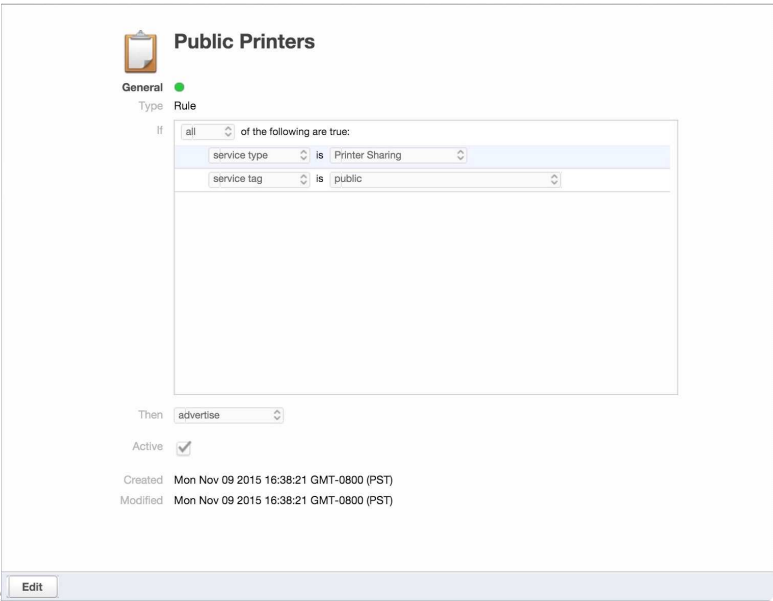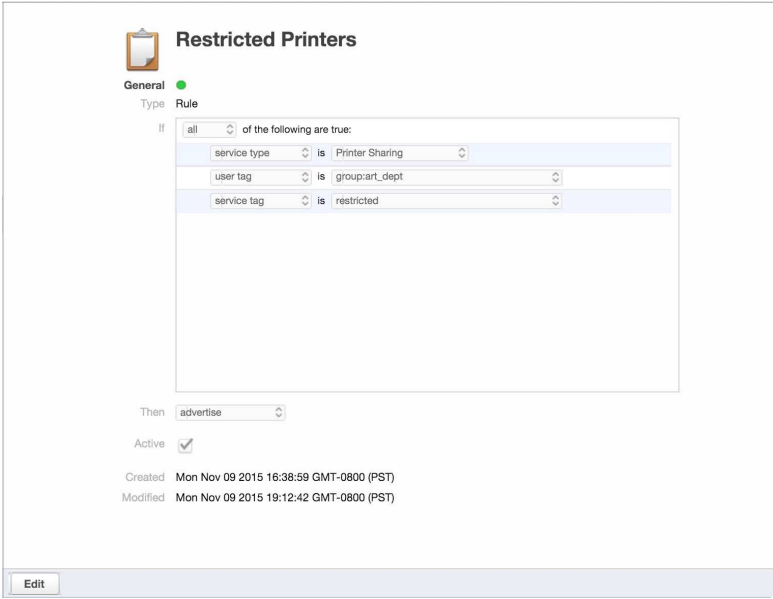
Figure 6: Public Printers Rule

Figure 7: Restricted Printers Rule

Simplify and Generalize

As simple as these rules are, they are still unnecessarily complicated. Since employees in the Art Department can print on any printer, it isn't necessary to tag the restricted printers. If we remove the service tag condition element from Restricted Printers, the rules will still work as desired. The edited, simplified Restricted Printers rule is shown in
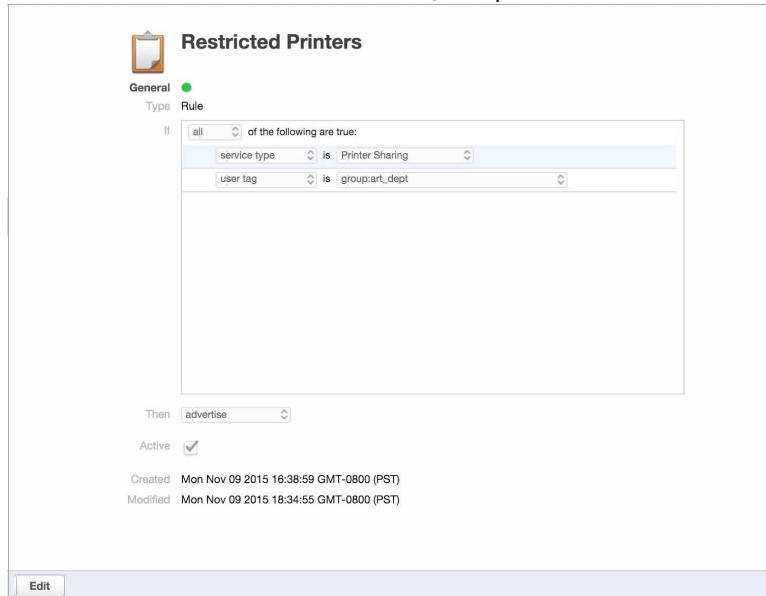

Figure 8.

Figure 8: Simplified Rule

Attaching the *restricted* tag to the Art Department's printers is also unnecessary. By simply omitting the *public* tag, we prevent the restricted printers from being advertised by the Public Printers rule.

This leads us to more Maxims:

> Maxim 4: Prefer simple rules (with fewer condition elements) to complex rules.

> Maxim 5: The best way to broaden a rule (that is, to increase the number of queries it advertises to) is to remove condition elements, not add them.

> Maxim 6: Make rules as general as possible (this is really just another way of stating Maxims 4 and 5 together).

More About Query Context

User tags are only available to rules if the user making the query is logged in to the Presto app on the mobile device in use, and the device supports the ability to send

context as part of the query message. At present, iOS devices cannot send user identity or location information in the query. The next section will explain how advertisement can be controlled if user identity and tags are not available.

Subnets

Every query includes a tag that identifies the subnet from which the query originated. The name of this tag is *subnet:* followed by the IP address of the subnet. This is always present in the query context, regardless of the type of device, or whether the user is logged in. Rules can be written to advertise to queries based on these subnet tags. This approach requires that users' devices are configured to connect to different wireless SSIDs, which are located on different subnets. Let's re-cast our previous example to work with this approach.

Scenario 2: Supporting AirPrint

For this scenario, our organization has two wireless networks with different SSIDs, that are on separate subnets—`192.168.0.0/24` and `192.168.1.0/24`. The mobile devices of the Art Department employees are configured to connect to the SSID on `192.168.0.0/24`, and other employees' devices are configured to connect to `192.168.1.0/24`.

With this approach it's easiest to think of a subnet tag as the rough equivalent of a user group tag, since you're using subnets to segregate user groups. It may seem a bit confusing, but the subnet tags appear to rules as if they are user tags, even though the subnet tag isn't part of the user profile. A more appropriate name for the condition element type might be *query tag*, since the tags are associated with the query, not necessarily with users. Nevertheless, use the user tag condition element to specify the subnet tags in a rule. The new version of the rule for Art Department printers (the *Subnet Restricted Printers* in Figure 9) specifies the appropriate subnet instead of the user group tag. Note that the Public Printers rule doesn't need to change, since it doesn't specify a group tag.
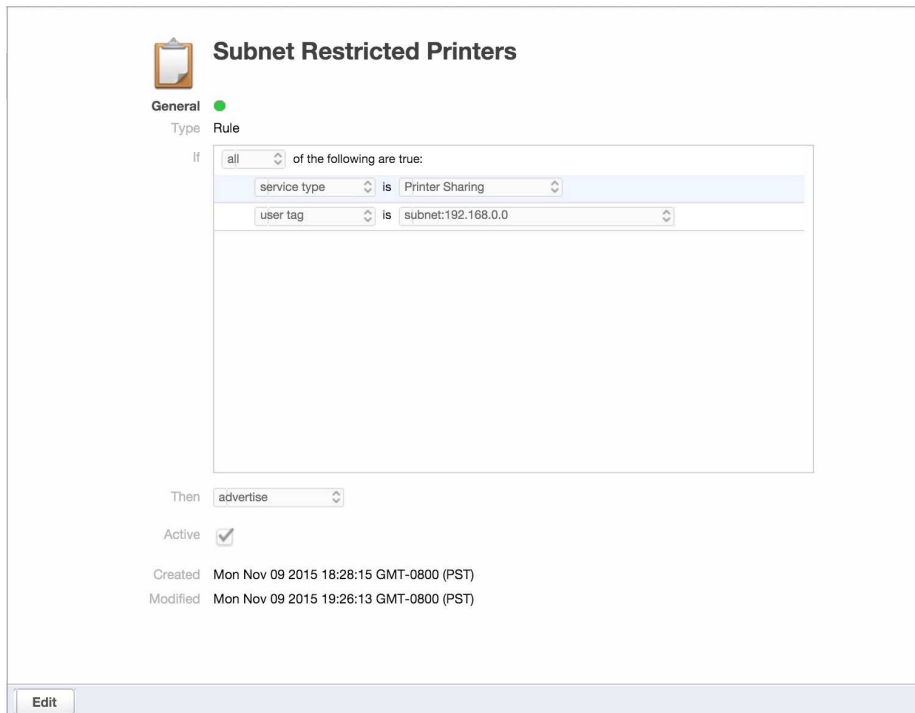
Figure 9: Using Subnet Tags in Place of User Group Tags

Combining Different Strategies

In an organization where users have a variety of devices (iOS, Android, Chrome OS), non-iOS devices support the inclusion of user ID and location. The original Public Printers rule will permit non-iOS device users in the Art Department to have access to their restricted printers, regardless of the subnet where they are located. Using Restricted Printers and Subnet Restricted Printers in the same configuration is a perfectly reasonable thing to do, and allows an administrator a little more flexibility in how devices and networks are configured.

It's worth noting that if an Android user in the Art Department were connected to subnet `192.168.0.0`, both rules would allow restricted printers to be advertised to that user. This is perfectly acceptable. It won't cause the printer to show up twice in the printer selection dialog.

Scenario 3: Using Compound Condition Elements

Now we'll expand the original scenario. Administrators have decided they want to be able to print to the fancy printers in the Art Department because…they are administrators, so who is going to stop them? We can extend the Restricted Printers rule to accommodate this. Edit the Restricted Printers rule as follows:

14

- Remove the condition element that specifies the user tag **group:art_dept** by clicking on the **–** (minus sign) at the right end of that condition element.

- Add a compound condition element to the rule by clicking the **+** (plus sign) on the last condition element in the rule while holding down the **Option** key.

- Select the **any** option in the **compound condition element** pop-up menu.

- Note that when the compound condition element is added, another condition element (by default, of type service name) is added beneath the compound element. Change this to a **user tag** condition element, and select the tag value **group:admin** from the **tag values** pop-up menu.

- Click the **+** (plus sign) on the right end of the user tag element you just added, to create another element. Change this element to a **user tag** element, and select the tag value **group:art_dept**.

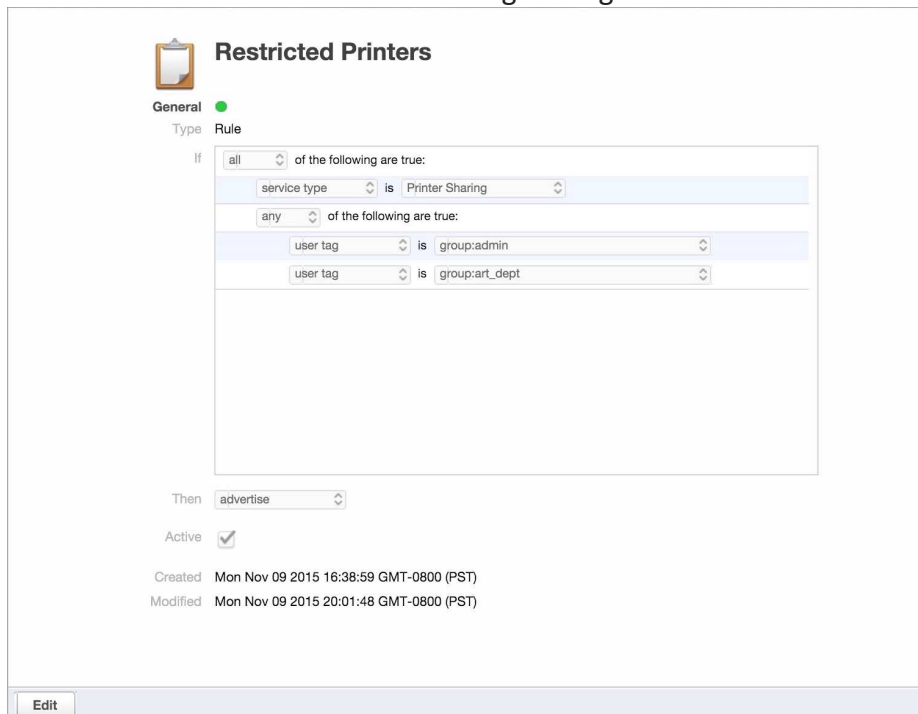- **Save** the rule. It should look something like Figure 10.

Figure 10: Compound Condition Element

The two user tag elements are indented beneath the compound element, indicating that they are subordinate to the compound element or to members of the group defined by the compound element. Because the compound element is of type *any*, it will evaluate to true if at least one of the sub-elements are true. While we're at it we can add the subnet tag for the subnet reserved for art_department employees with iOS devices and eliminate the rule Subnet Restricted Printers (see Figure 11).
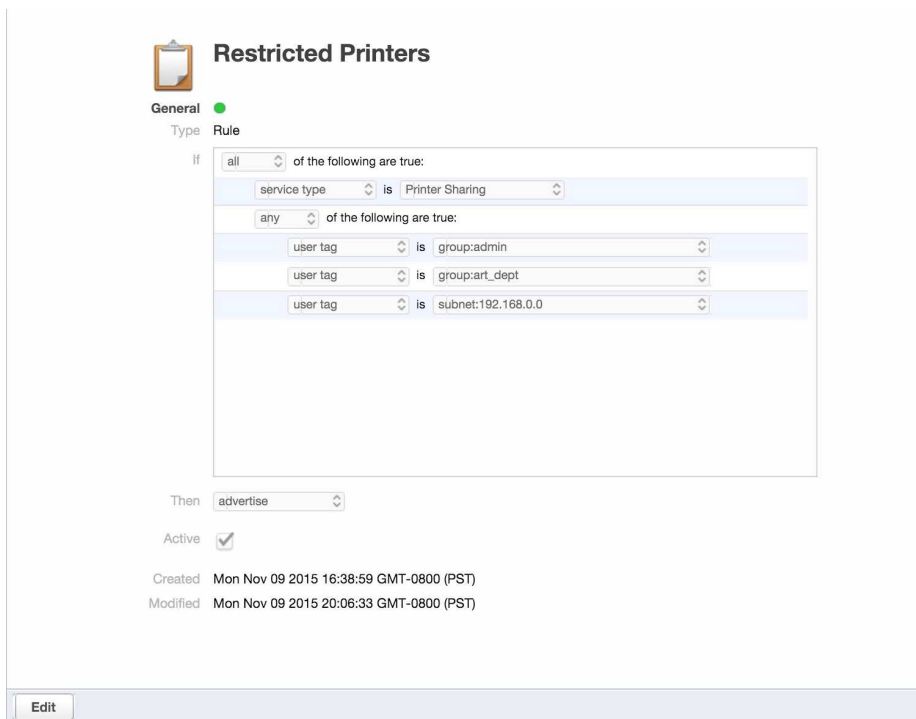
15

Figure 11: Adding Sub-elements to the Compound Element

Any query that satisfies at least one of these three elements will have all available printers (public or restricted) advertised in its response.

Loose Ends

There are a few topics that haven't been covered by these scenarios that still merit some discussion, including mDNS and rules prohibiting advertisement.

What is mDNS All About?

Presto depends on DNS services to perform its job. Most medium or large enterprises have internal DNS servers that can be configured to forward appropriate requests to the Presto server. In contrast, small organizations typically use external DNS servers (provided by their ISPs or publicly-available servers like OpenDNS), consisting of a single subnet. The configuration of DNS services for Presto installations is covered in detail in the *Presto Administrator Guide*.

In small environments consisting of a single subnet with external DNS servers, Presto relies on multicast DNS queries to support iOS devices. The Printer Sharing (mDNS) rule exists to support such small environments. By default, this rule is not active. If your network environment fits this description, set the Printer Sharing (mDNS) rule to **active**. Otherwise (if you have internal DNS servers configured to support Presto), leave this rule inactive. Do not edit the condition of this rule.

Rules that Prohibit Advertisement

If the action of a rule is set to *do not advertise*, and one or more query/service pairs satisfy the rule's condition, that rule will prohibit the services from being advertised to the queries with which they are paired. A rule that prohibits advertisement takes priority over rules that might propose advertising the services to that query. In general, prohibitive rules are a bit like sledgehammers—they are powerful, but must be used with care. For best practices, if you can achieve the desired behavior using only rules that advertise, do so.

Summary

Presto's rules facility provides unprecedented control and flexibility over service discovery and access in Enterprise environments. As we continue to improve our product and work with our customers to understand and solve their problems, expect to see new capabilities added—more context to reason about, new condition elements, richer logical expressions—informed by our customers' experiences and needs. We will continue adding to and refining this document, so that over time it evolves into a true guide to best practices for Presto rules.