**ibm ICD**

**INTEGRATION with**

**jira software**

Contents

[RevisionHistory 3](#_Toc69208403)

[REQUIREMENT DESCRIPTION 4](#_Toc69208404)

[Solution Description 4](#_Toc69208405)

[Installation of JIRA Server 7](#_Toc69208406)

[Project creation in JIRA 7](#_Toc69208407)

[Generation of Auth token in JIRA 8](#_Toc69208408)

[Installation of Automation Rule plugin in JIRA 9](#_Toc69208409)

[Creation of automation rules to add webhook in JIRA 9](#_Toc69208410)

[How to Add custom fields in JIRA 12](#_Toc69208411)

[MAP JIRA TICKET STATUS TO ALIGN IT with ICD 15](#_Toc69208412)

[ADD JIRA ENDPOINTS IN ICD 16](#_Toc69208413)

[ADD DOMAIN IN ICD 17](#_Toc69208414)

[Add new attributes in Incident application 18](#_Toc69208415)

[Create an automation script to create incident in jira 18](#_Toc69208416)

[Create an automation script to update incident 20](#_Toc69208417)

[EXECUTION WITH MAXADMIN 20](#_Toc69208418)

[Incident creation using different user 23](#_Toc69208419)

# RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision #** | **Revision Date** | **Description of Change** | **Author** |
| 0.1 | 08-Apr-2021 | Initial Version | Shubhi Jain |

# REQUIREMENT DESCRIPTION

ICD is used to provide users ability to raise tickets for IT services, including issues on Software.

In-house Software teams use Jira to manage Issues/Projects. They generally prefer the Atlassian suite because it is used for Coding/Collaboration

Using ICD capability, JIRA is integrated to manage projects and issues.

As part of this POC, we have covered below use cases-

1. **Create issues in Jira:** Incidents reported by a user on ICD need to be forwarded to Jira as an issue to be solved by the Software team
2. **Update in Issue:** The status of the issue from Jira needs to be reflected in the incident ticket. Also, update in Description, priority of the issue will be reflected in ICD incident.
3. **Worklog update:**Any update in the comment in JIRA defect, will be updated as a worklog in ICD incident.

# Solution Description

IBM ICD is a highly configurable product. With minimal technical skills, users can add new fields, make screen changes, or restrict values of attributes in ICD.

**Leveraging the power of the below ICD solution blocks, ICD is integrated with JIRA-**

1. [**Automation Scripts**](https://www.ibm.com/docs/en/maximo-for-utilities/7.6.0?topic=SSLLAM_7.6.0/com.ibm.mbs.doc/autoscript/c_automation_scripts.html): Automation scripts are small, targeted pieces of code that can extend the product. With the Automation Scripts application, you can create scripts to automate tasks based on the events or attributes of a business object or based on actions or custom conditions.

The biggest advantage in Automation Scripting is the deployment process. Unlike Java customization that can take hours to deploy if you’re clustered, which makes developing and testing take substantially longer, Automation Scripts can be added, modified, or disabled without any downtime. That alone makes it hard to beat for most organizations.

**Components of automation scripts**

For an automation script to run, you specify the following components:

* **A launch point** that defines the context for the script to run - A launch point defines the execution context for an automation script, for example when a business object is updated, or a value is entered into a field. Some types of launch point are supported, and the Automation Scripts application provides a separate wizard application for creating each type of launch point.
* **Variables and corresponding binding values** - You can specify the variables that determine how information is passed into or received from an automation script in the wizard applications. Variables are not mandatory, but when you use variables it simplifies the amount of code that is written and makes it easier to reuse the code.
* **Source code** written in a supported scripting language - You can write source code in an external application and import it into the Automation Scripts application.

1. [**Endpoint Configuration**](https://www.ibm.com/docs/en/cdfsp/7.6.0?topic=components-endpoints-handlers)**:** An endpoint and its associated handler routes outbound messages to an external system. The combination of an endpoint and handler specifies the transport protocol to use, such as HTTP or web service, and provides the communication data required to reach the destination, such as a URL.

You create an endpoint to identify a target location and the transport mechanism that the integration framework, or deployment manager uses to publish data, or to invoke a service.

You can add a handler to an endpoint record to specify how to route outbound data to a specific endpoint location. You can also add a handler to define the data format that is used in the data transfers.

For JIRA Integration, HTTP Handler is used by providing the metadata information.

1. [**Application Designer**](https://www.ibm.com/docs/en/SSLKT6_7.6.0/com.ibm.mbs.doc/designer/c_application_designer.html): The Application Designer tool helps the user create the XML for each application, without touching the XML.

Applications in IBM ICD are stored as an XML document in the database. This XML is then rendered by Maximo to show the different applications that are available. A palette with all the available elements is available and to add the elements to the application it is just drag and drop to the desired place. Apart from the application itself, one can also choose to edit the dialogues related to the application, signature options, toolbars and select action menu.

The Control Palette in the designer has the whole list of elements which can be added to any application. The application will be split in sections, and then, in the sections one can add Tables, text boxes, buttons, Images, Hyperlinks, and other elements. After adding the elements, themselves, one will also need to change the properties of each element.

1. [**Database configuration**](https://www.ibm.com/docs/en/cdfsp/7.6.0?topic=module-database-configuration-application)**:** You use the Database Configuration application to create or change objects and attributes, and to customize the database. An object is a self-contained software entity that consists of both data and functions to manipulate data. Most applications are associated with one main object and other related objects.

When you use the Database Configuration application, you interact at the business object level. Internally, the application determines the actions to take on the database tables to support the needs of business objects.

Adding/Editing Attributes: Apart from adding MBOs, you can add Attributes to existing MBOs. Attributes can also be Non-Persistent, even if the parent MBO is itself persistent. In the Attribute section, one would set the type of the attribute, default value, class, domain and many more features.

Although domains are a separate application, they are also used with attributes. Setting domains to attributes will help you show and validate the attribute data against a specific set of data. From the database configuration, you can also set default values to attributes, set auto-numbering and set if an attribute is required or not.

**Above building blocks of ICD supported JIRA Integration with ICD without doing in any Java level code changes. There is no need to re-deploy ICD after making these changes. Hence, without any downtime we will be able to achieve our goal.**

Below are the steps that needs to be performed for the integration-

1. Jira should be configured with API Token for authentication
2. Automation Rules plugin should be enabled for webhook configuration. This is required for the two-way communication with ICD. Whenever there is any update in JIRA ticket, same can immediately updated in ICD. Rules should be imported with the required JSON attributes.
3. Required custom fields should be configured in JIRA for to way mapping with ICD.
4. JIRA Endpoint should be added in Endpoints application.
5. Add a new Domain for Endpoints in ICD for multiple JIRA instances.
6. In Database Configuration, two new attributes Create jira ticket (Checkbox whether user wants to create ticket in jira) and Jira Endpoint (JIRA Server url) are added in ICD.
7. Incident application should be customized with the new attributes using Application Designer.
8. JIRA workflow should be mapped to ICD workflow and different fields like priority, status, description is mapped according to the JIRA fields.
9. Automation Scripts should be added with proper launch points.
10. Please note- There was jar conflict for httpcore jar in TPAE folder as it had 2 versions of the same jar and our automation script was giving ClassNotFoundException. As temporary fix we have removed the “httpcore-4.2.4” and using the latest one “httpcore-4.4.4”in TPAE lib folder.

 CREATE BUG IN JIRA (ENDPOINT PRESENT)



UPDATE DESCRIPTION, STATUS, PRIORITY

# Installation of JIRA Server

1. Install Jira Server

Use below link to install jira server according to your machine configuration

Windows - <https://confluence.atlassian.com/adminjiraserver/installing-jira-applications-on-windows-938846835.html>

Linux- <https://confluence.atlassian.com/adminjiraserver/installing-jira-applications-on-linux-938846841.html>

1. Once jira server is installed, your jira should be accessible from a URL like-

http://<computer\_name\_or\_IP\_address>:<port>

eg- <http://172.20.152.18:8080>

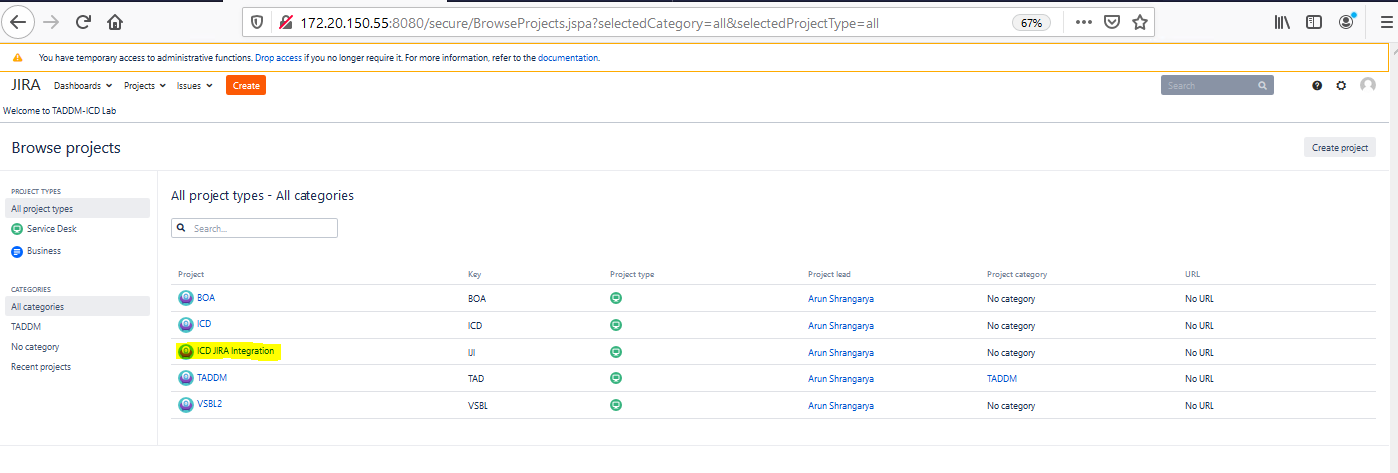
1. Login using Administrator, you can add few more users and invite your team members.

Go to Settings- User Management – on right click on create new user.

Create a new user with required role (you can do this without SMTP server as well).

# Project creation in JIRA

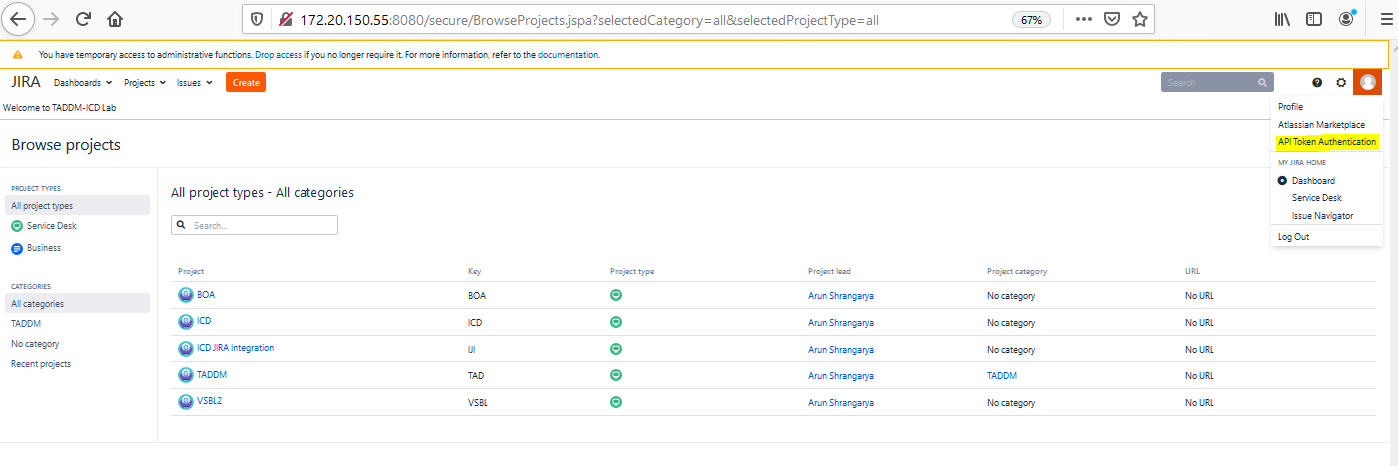
1. Create a project in JIRA to store ICD tickets, please note it’s project key for API call like in the below image



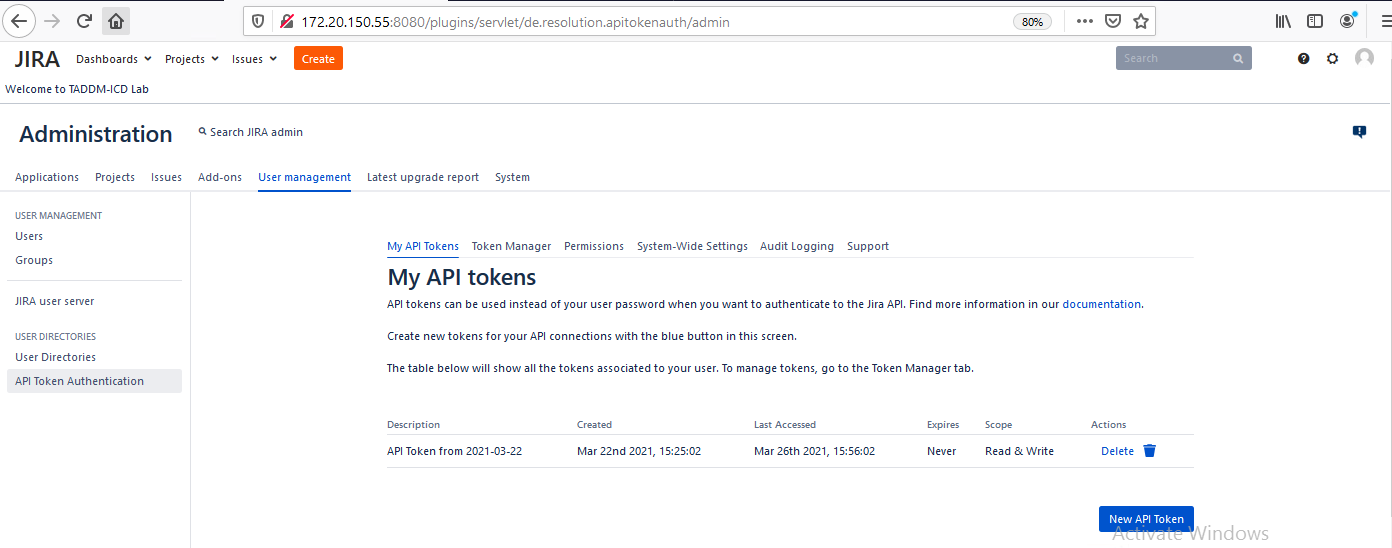
1. We will link Incidents applications in ICD to JIRA defects.

# Generation of Auth token in JIRA

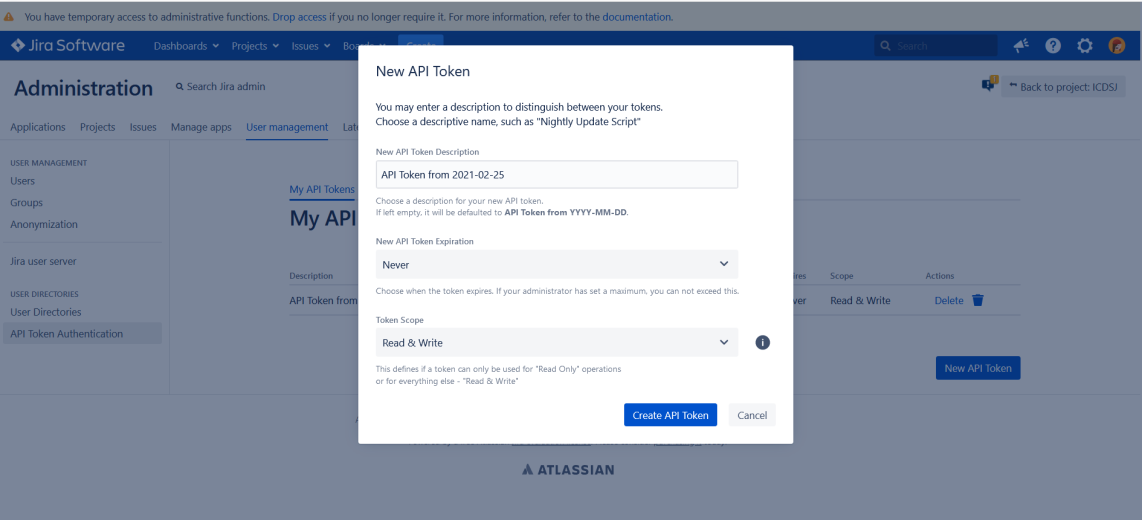
1. Create an API token in JIRA to integrate it with external system i.e ICD in our case.
2. Click on Profile icon on top right-> you can see API token Authentication, click on it.



1. Click on New API Token



1. Provide name, expiration and scope to token



1. Click on create API token, copy the token and keep it with you. This token will be used as part of Basic Authentication password in ICD.

# Installation of Automation Rule plugin in JIRA

1. Install Automation Rules Plugin in JIRA server.

Automation rules are used to perform actions in your service project based on specific triggers and conditions. For example, you can set an automation rule that alerts an agent when a high priority issue is created. Or, service project can reopen an issue if your customer comments on it after its been resolved.

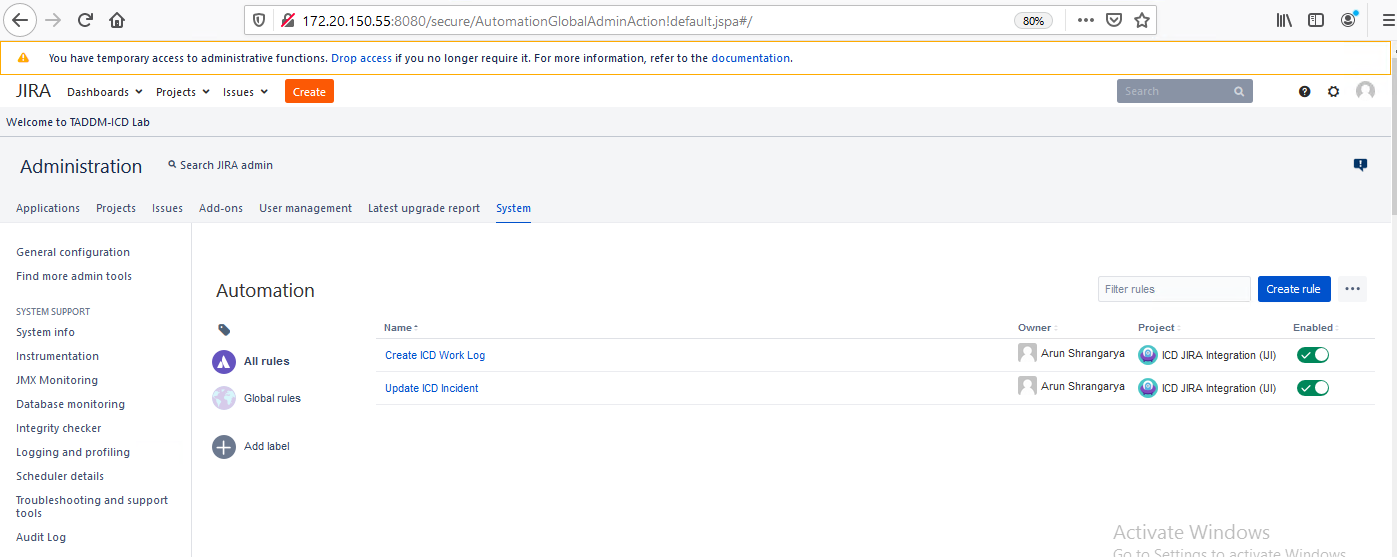
1. Go to Settings- > Manage apps
2. Search for Automation Rules in marketplace
3. Install it.

# Creation of automation rules to add webhook in JIRA

1. Go to Settings-> System -> Automation Rules, to create a rule for webhook.

Webhook is basically used to communicate to ICD from JIRA on any event in issue in JIRA.

For eg- If user has updated the issue summary, by configuring webhook we can trigger the external system and process the information.

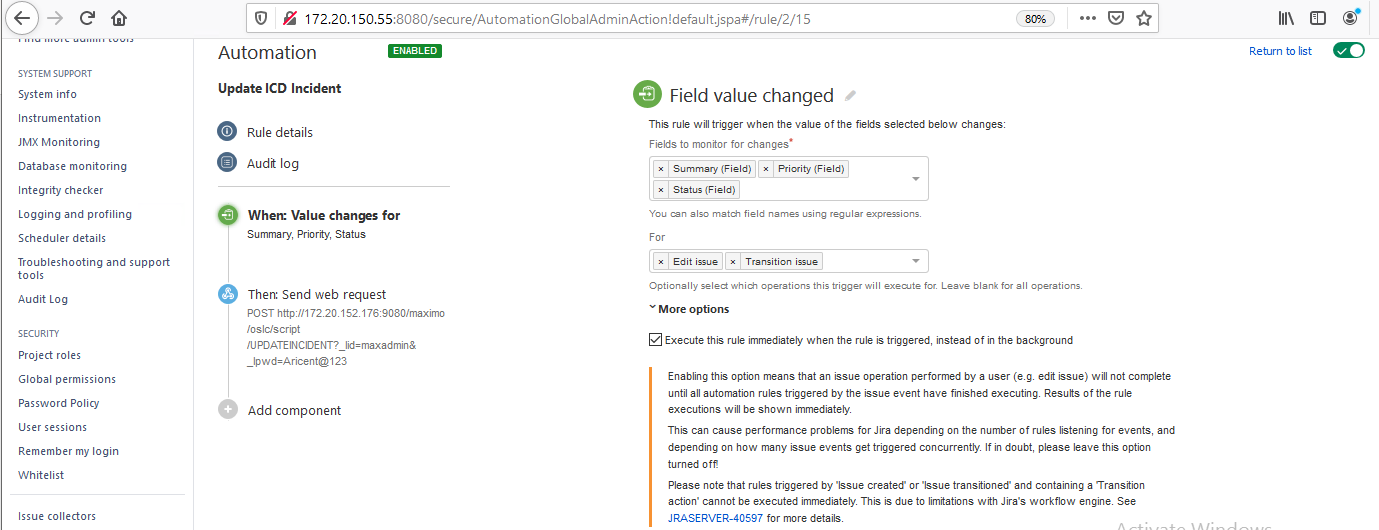


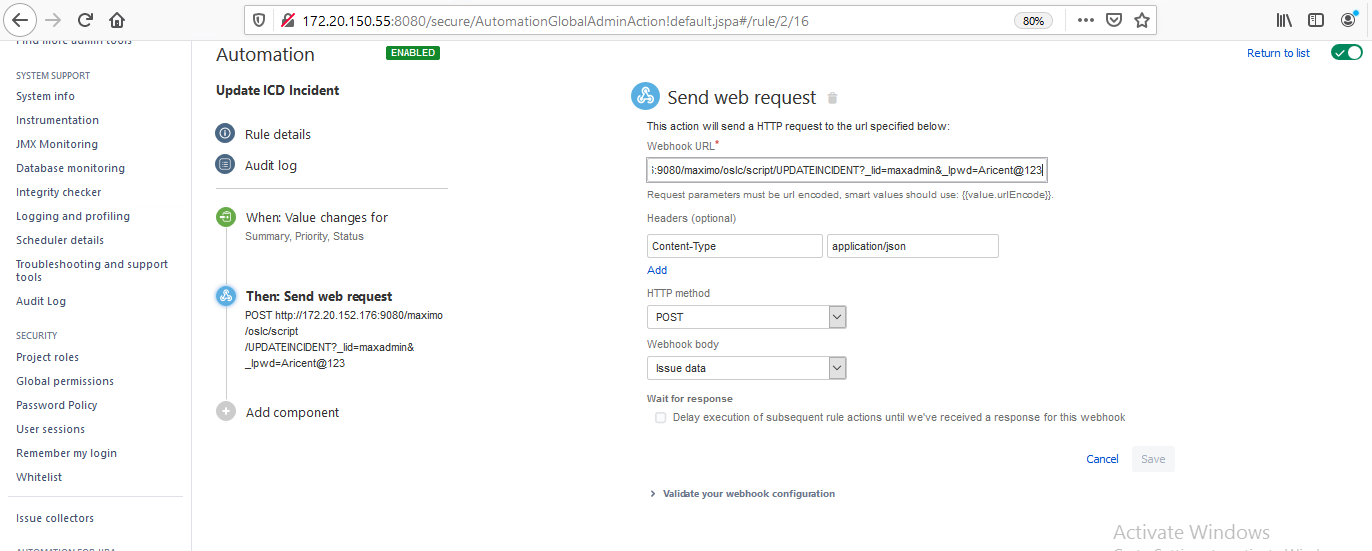
1. Click on create rule to add a new webhook.

Follow below link to create a rule and send the data in your webhook request-

<https://www.atlassian.com/software/jira/guides/expand-jira/automation>

1. Example –

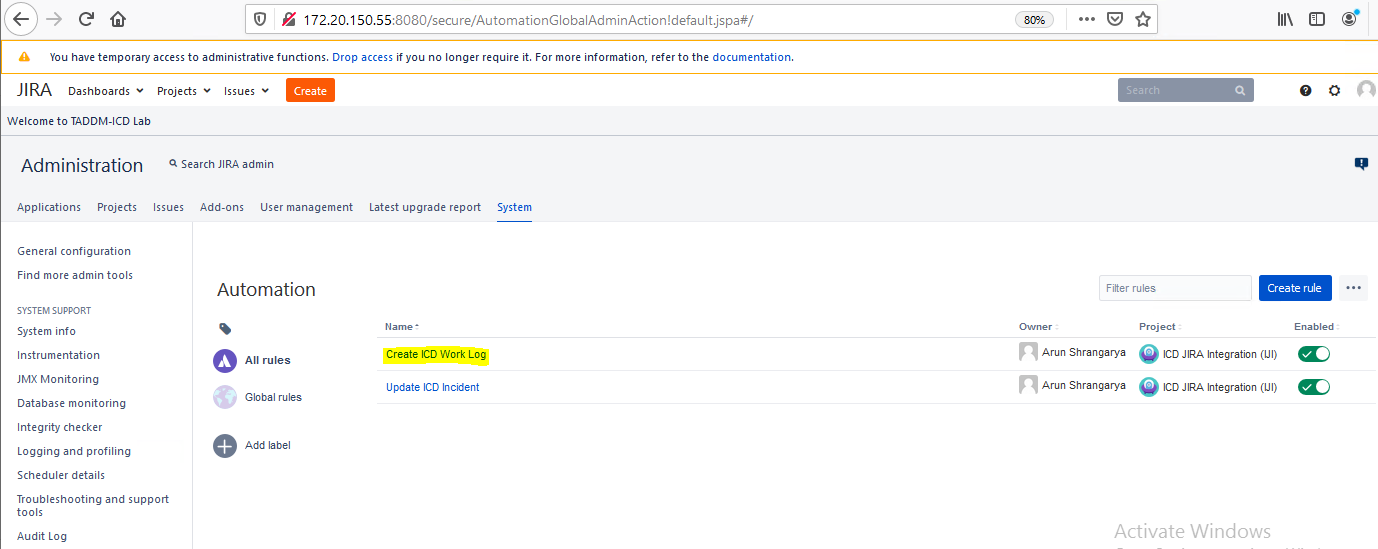


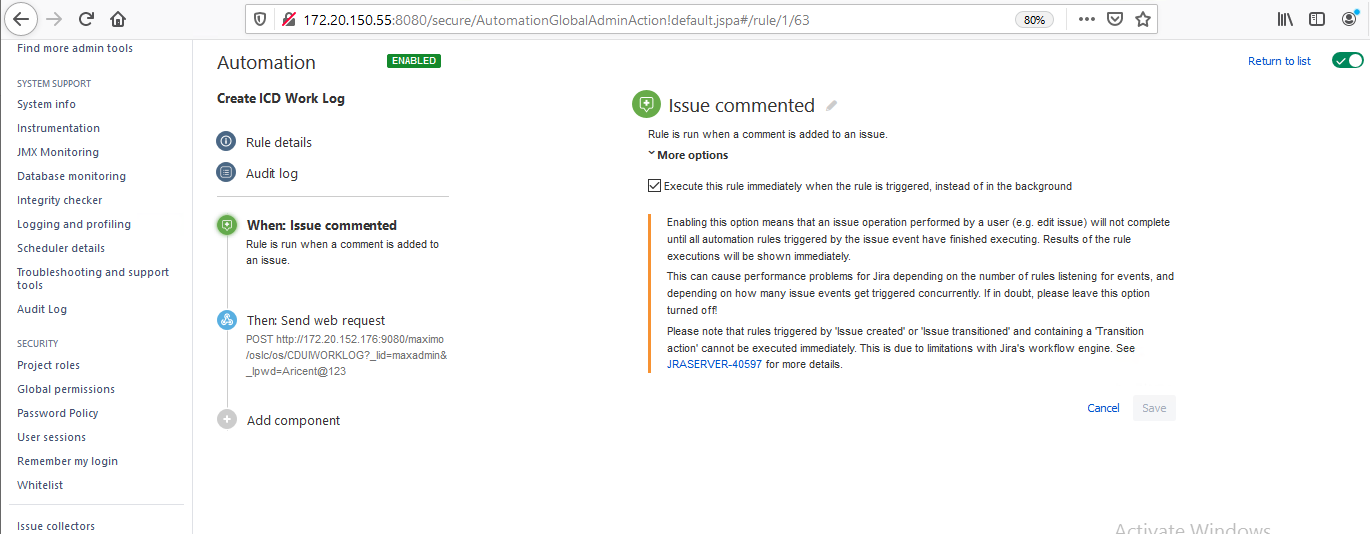


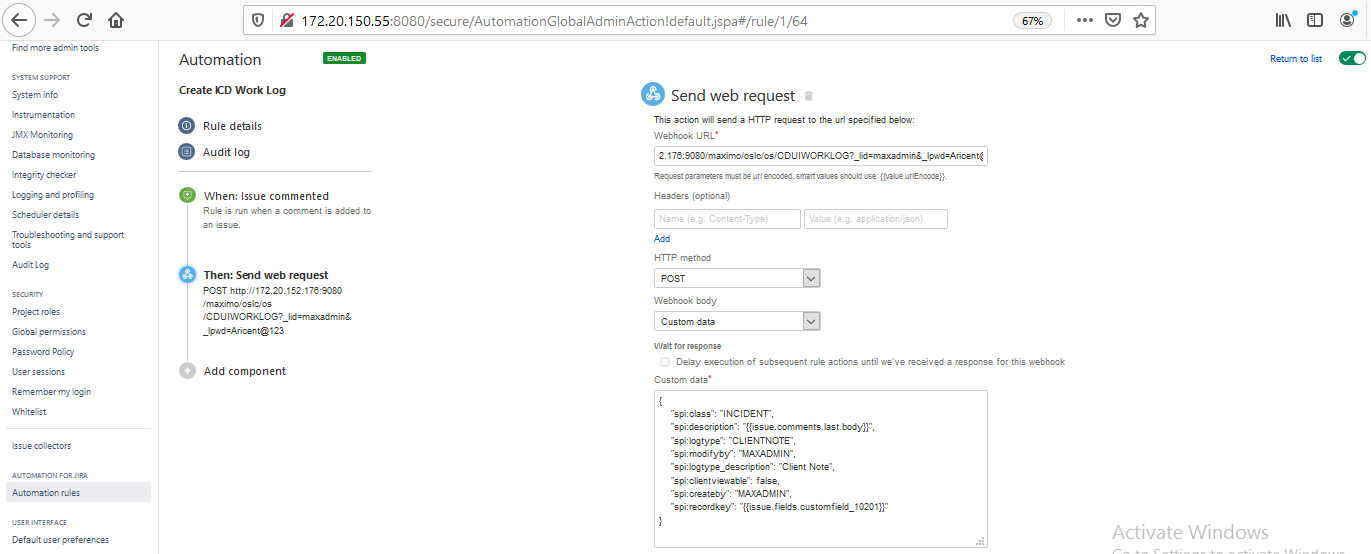
* In above example we have added ICD automation script as our webhook url-
* <http://172.20.152.176:9080/maximo/oslc/script/UPDATEINCIDENT?_lid=maxadmin&_lpwd=Aricent@123>
* You can validate your webhook end point by providing issue id and validate button and check the audit logs for the response.
* Here we have selected full issue details to be sent with the request to capture and save all the details changes back in ICD.
* Publish your rule.

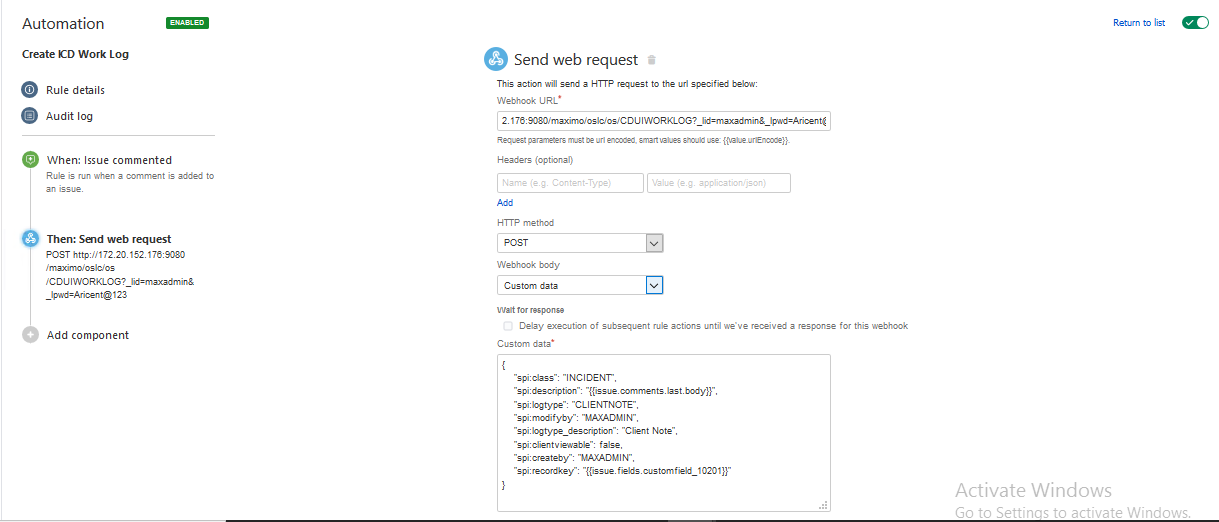
Below attachment is for WorkLog automation rule in JIRA. Directly import below json file inside JIRA Automation Rules to create work logs from JIRA to ICD











For Worklog synchronization, in the automation rule, we need to pass custom JSON data that is required by ICD API to update worklog in ICD. Check above screenshot for reference.

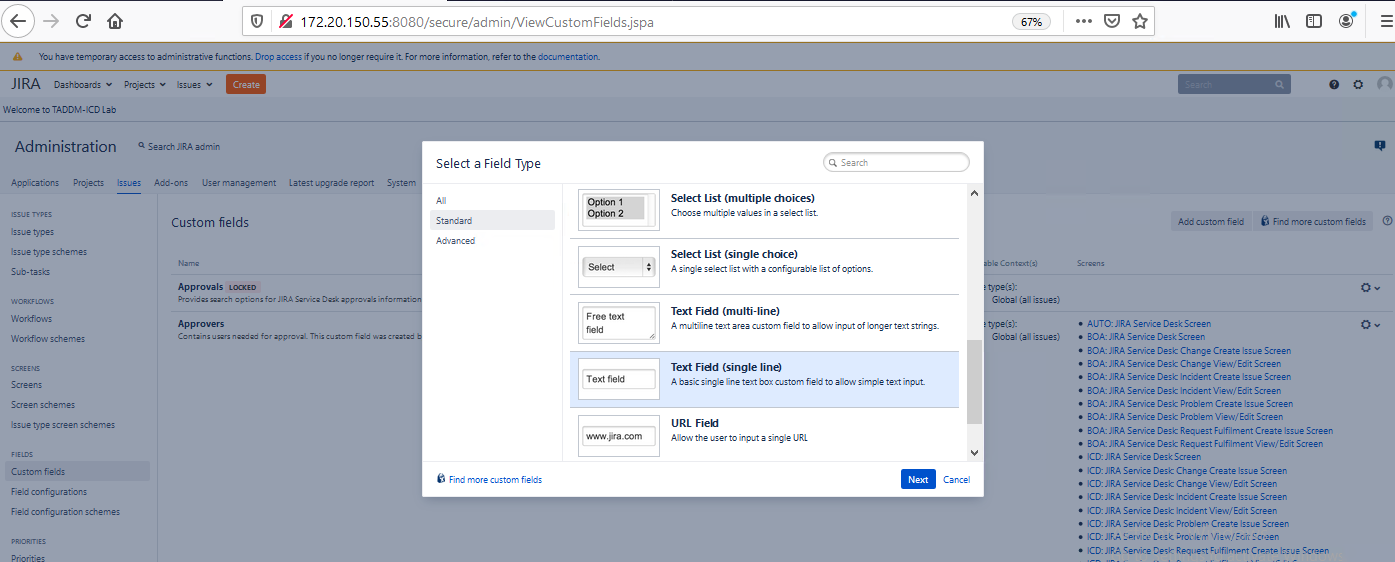
# How to Add custom fields in JIRA

1. For this use case we need to create two custom fields for ICD to jira mapping.

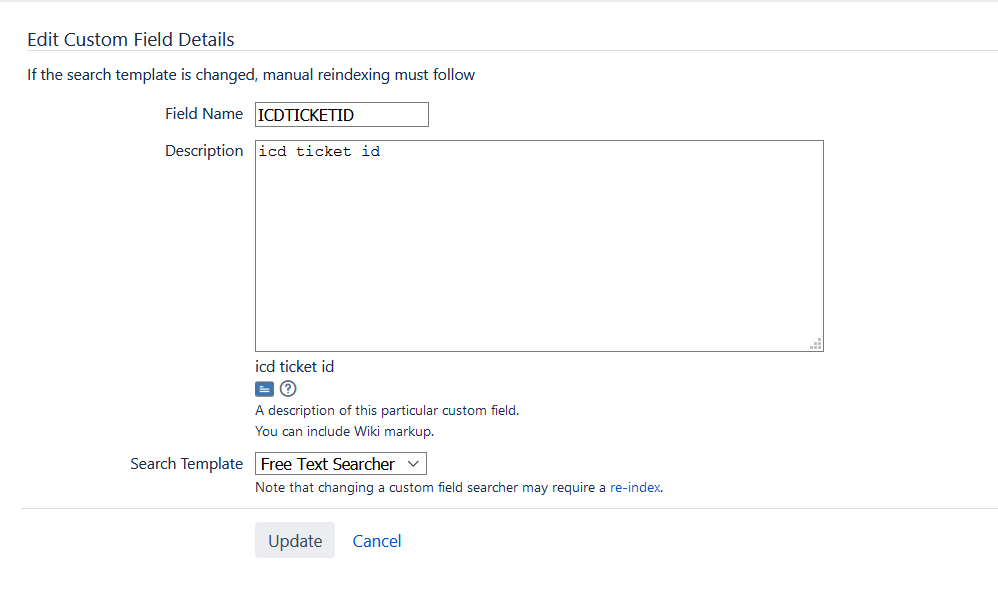
In ICD, ticketuid is unique key for tickets, so we are storing ticketuid in jira custom field (ICDTICKETID).

Also, create direct ICD Incident ID in JIRA for worklog management.

1. To create a custom field, go to settings-> issues -> Fields category - > custom field
2. From Top right, click on create custom field, choose the type of field. Here we have selected a textbox.

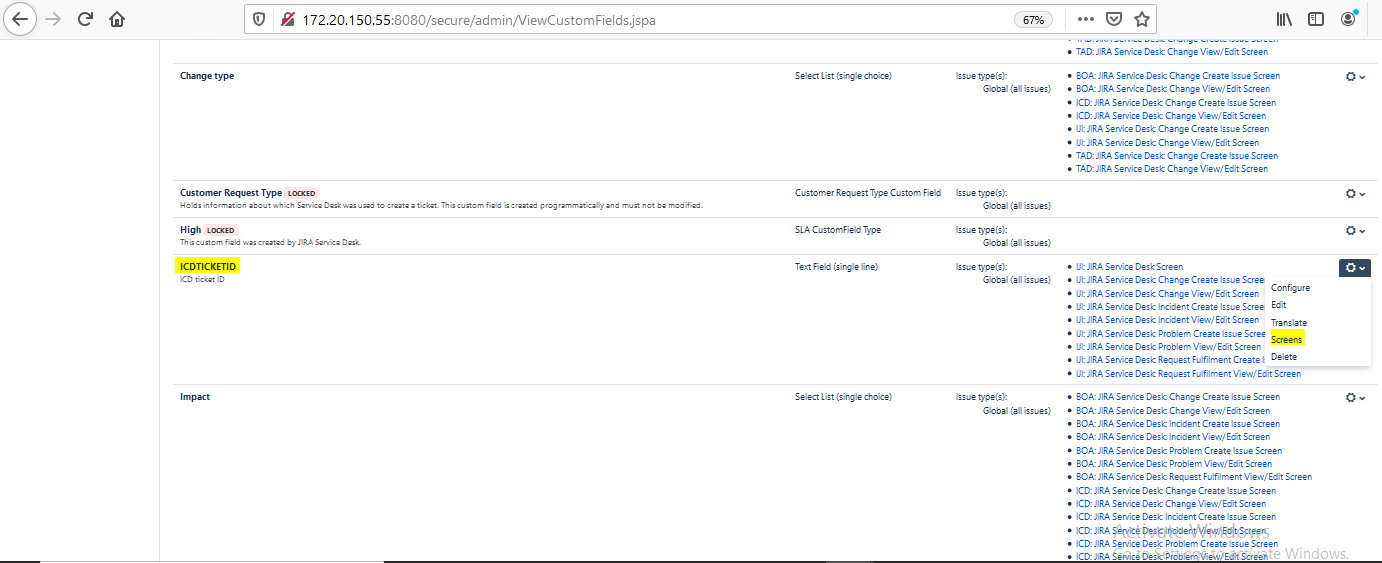


1. Add name, description for the field

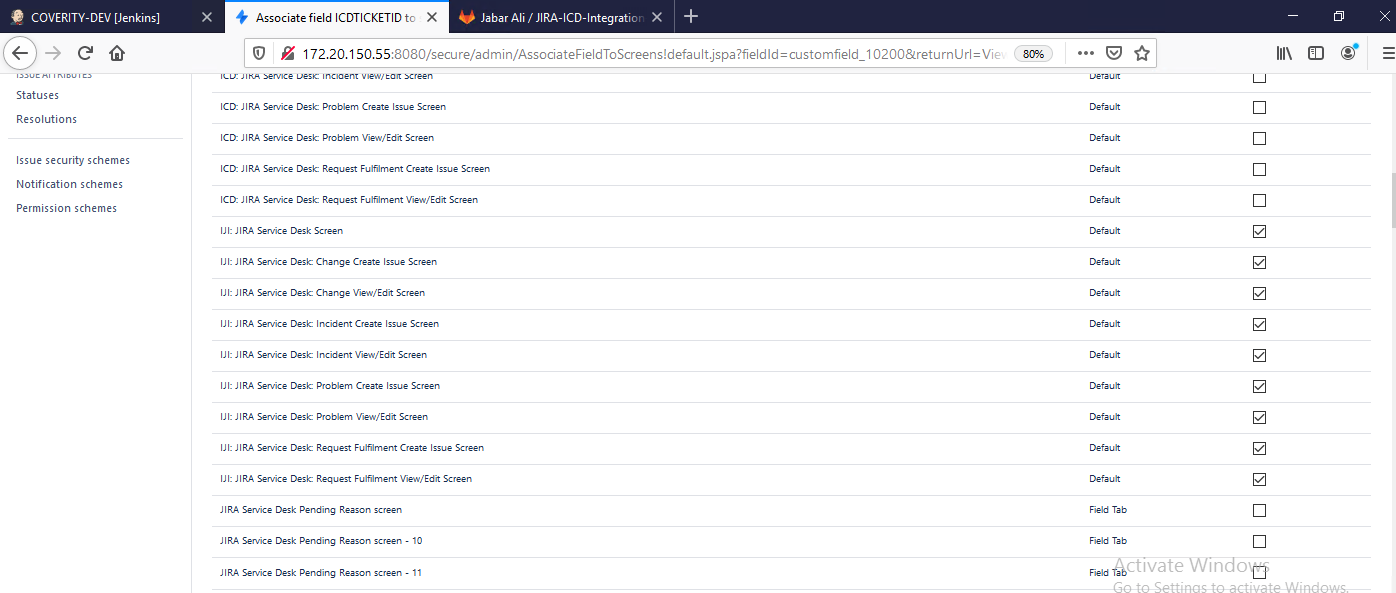


1. Customize the fields according to the requirement

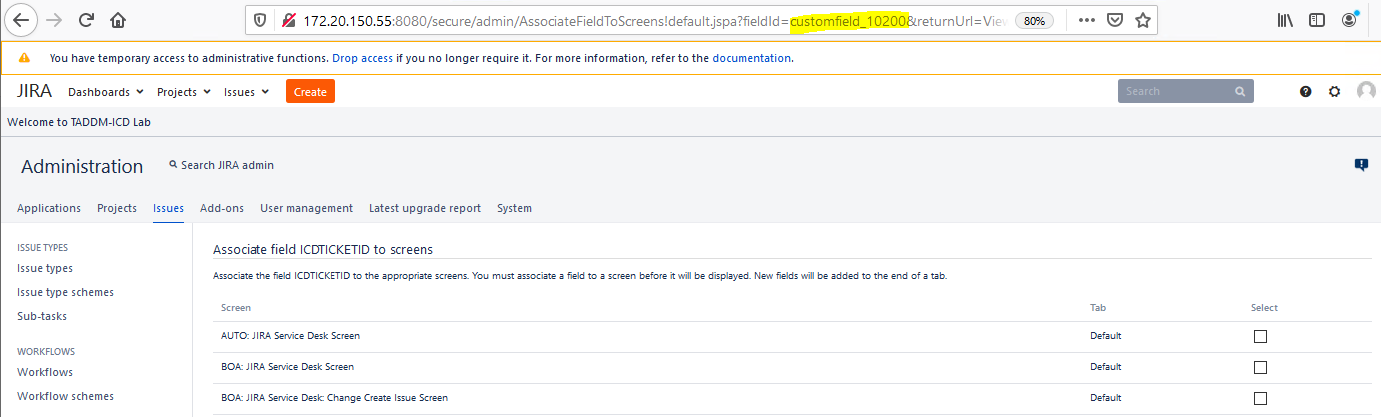
For this, Go to Custom Field, select the settings symbol next to the newly created field, go to screens options.



1. Select the checkbox for which screen you want this field to get displayed and update it.



1. To get the field, key check the url of the same webpage and copy it for reference.



Eg- in our case it customfield\_10200

# MAP JIRA TICKET STATUS TO ALIGN IT with ICD

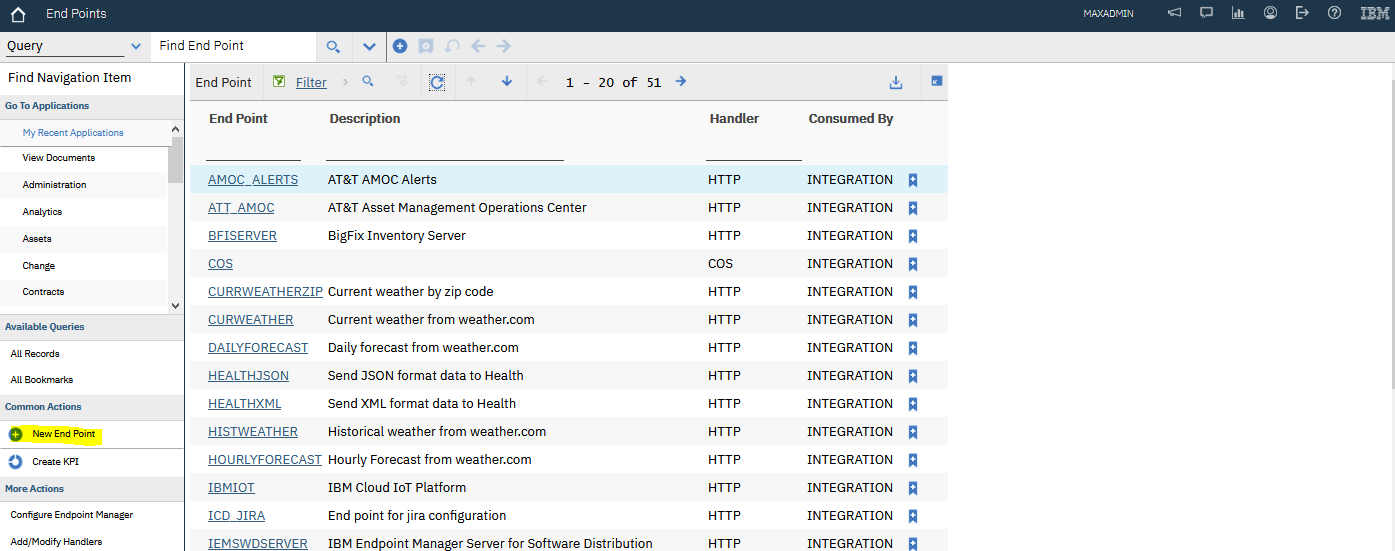
**Error! Not a valid embedded object.**

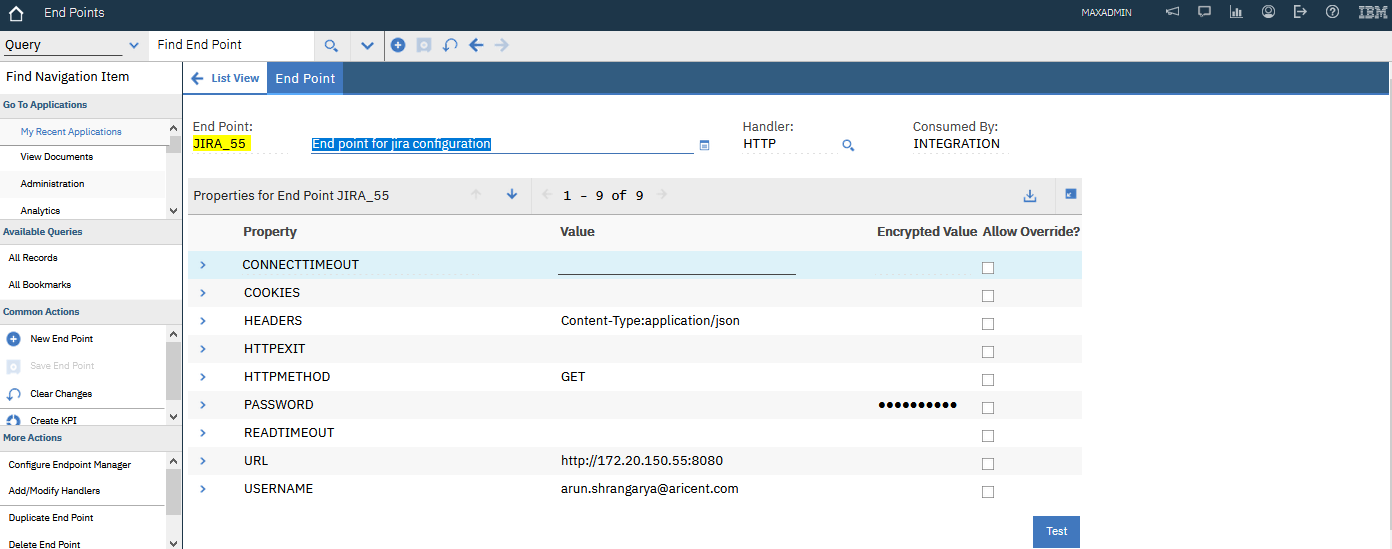
# ADD JIRA ENDPOINTS IN ICD

[Endpoint](https://www.ibm.com/docs/en/cdfsp/7.6.0?topic=endpoint-creating-endpoints) is created to identify a target location and the transport mechanism that the integration framework, or deployment manager uses to publish data, or to invoke a service.

1. Create End Points to add details of the JIRA server.

Go to End Points Application, create a new Endpoint and add HTTP Handler. Provide url, username, password and headers



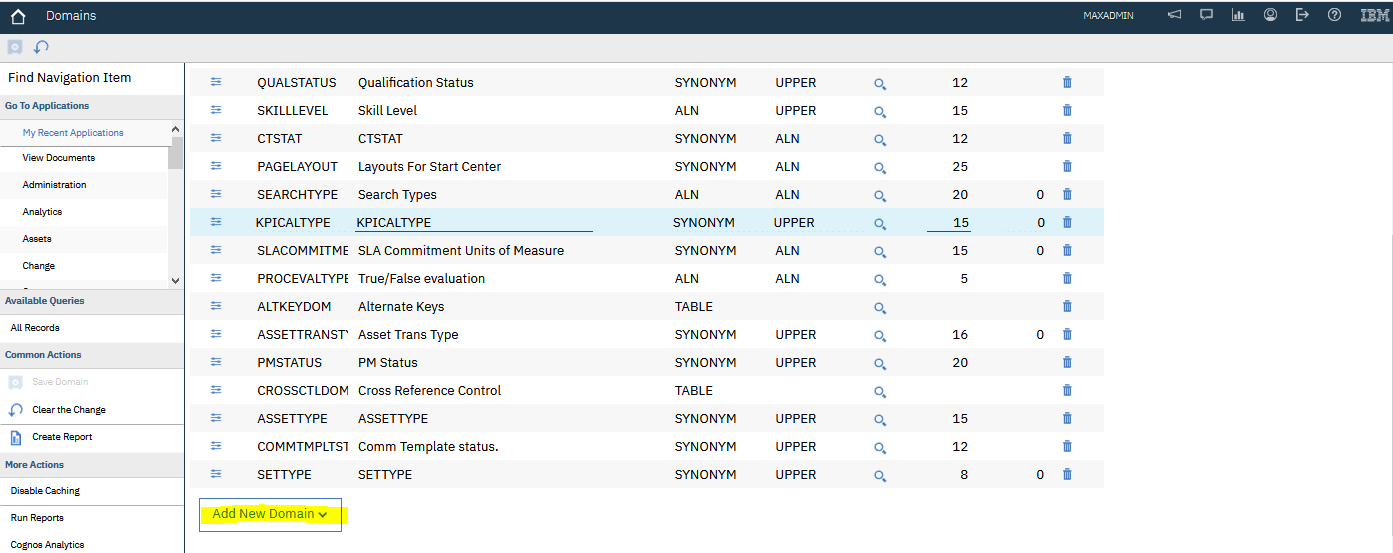


# ADD DOMAIN IN ICD

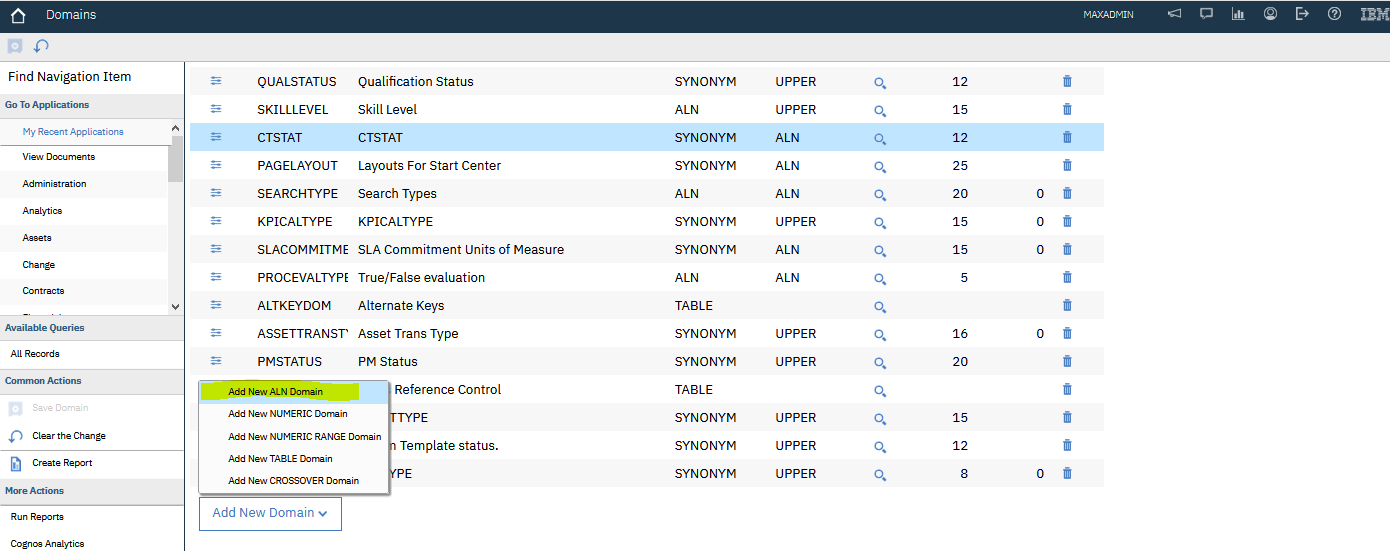
[Domains](https://www.ibm.com/docs/en/maximo-eam-saas?topic=overview-types-domains) define the values that are allowed in fields. They can be available in lookup list dialog boxes that are also referred to as Select Value lists. Domains also define the values that can be typed directly in a field. Domains helps ensure that valid and consistent data is entered and reported on. Domains are used throughout the product.

1. Create Domain to configure Endpoint –

Go to Domains application



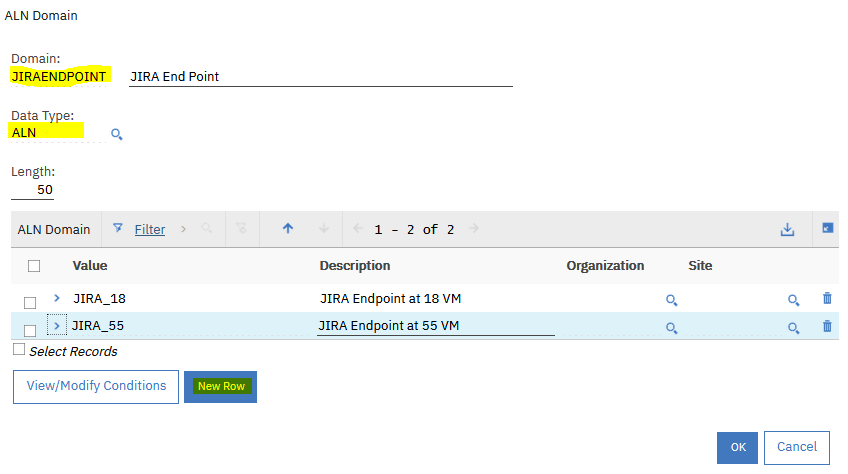
1. Add New ALN domain ->



1. Provide Domain name – JIRAENDPOINT

Configure Endpoints i.e JIRA Server details

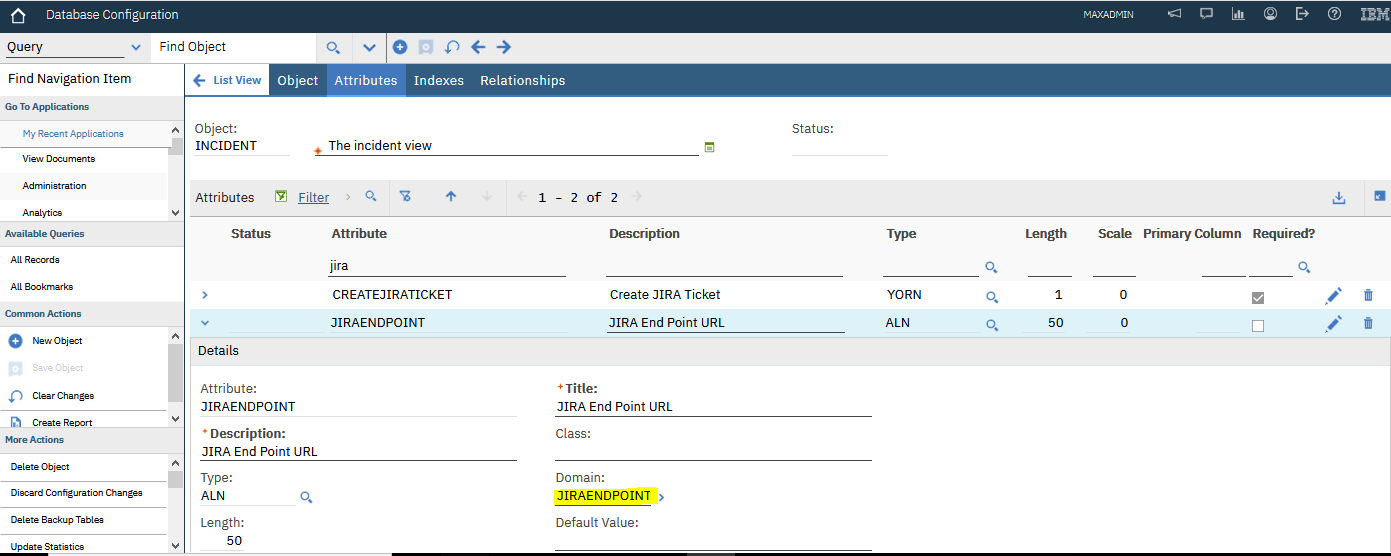
Eg- JIRA\_18 and JIRA\_55



# Add new attributes in Incident application

1. Configure domains created above to Incident application.

Go to Database Configuration application, search for Incident Object Incident and Add JiraEndpoint in the attributes.



# Create an automation script to create incident in jira

An [automation script](https://www.ibm.com/docs/en/maximo-for-utilities/7.6.0?topic=SSLLAM_7.6.0/com.ibm.mbs.doc/autoscript/c_automation_scripts.html) consists of a launch point, variables with corresponding binding values, and the source code. You use wizards to create the components of an automation script. You create scripts and launch points or you create a launch point and associate the launch point with an existing script.

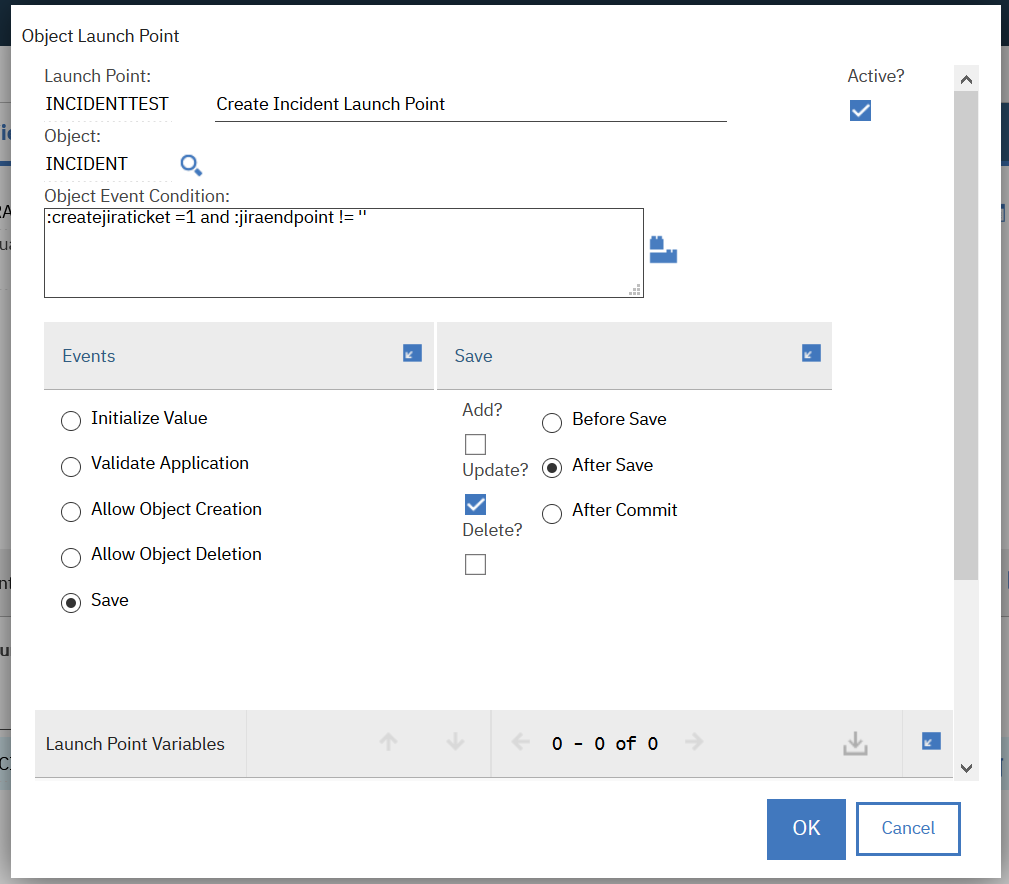
We need to create two automation scripts-

1. To create a jira issue whenever any ICD incident is raised based on input provided.
2. To update the ICD incident whenever any jira issue is updated

# Create an automation script to JIRA Tickets

**STEPS-**

1. Go to Automation script application, from more actions click on create – script with object launch point.
2. ! name as incident and fill the details as shown in the below image-



1. Select script name as jython, select it as active and log level as debug to get debugging logs. Enter the script content as in attached text file

**Error! Not a valid embedded object.**

# Create an automation script to update incident

1. Similarly create an automation script to update incident for webhook support from jira.

Use the below text file for reference.



1. Use your jira username as username and API token as password for basic authentication

Eg- auth = 'shubhijain' + ":" + 'pThwBjJFAaAKmkxPSLH5qI3Qd3XLqt7njb32oH'

1. Use your jira server IP:port as your jiraurl: [http://172.20.150.55:8080/rest/api/2/issue](http://172.20.152.18:8080/rest/api/2/issue)

# EXECUTION WITH MAXADMIN

1. Create an incident in ICD with Summary and basic details, JIRA ticket will be created automatically with the same Summary. Also, ICD ticketuid will be mapped as ICDTICKETID field in jira

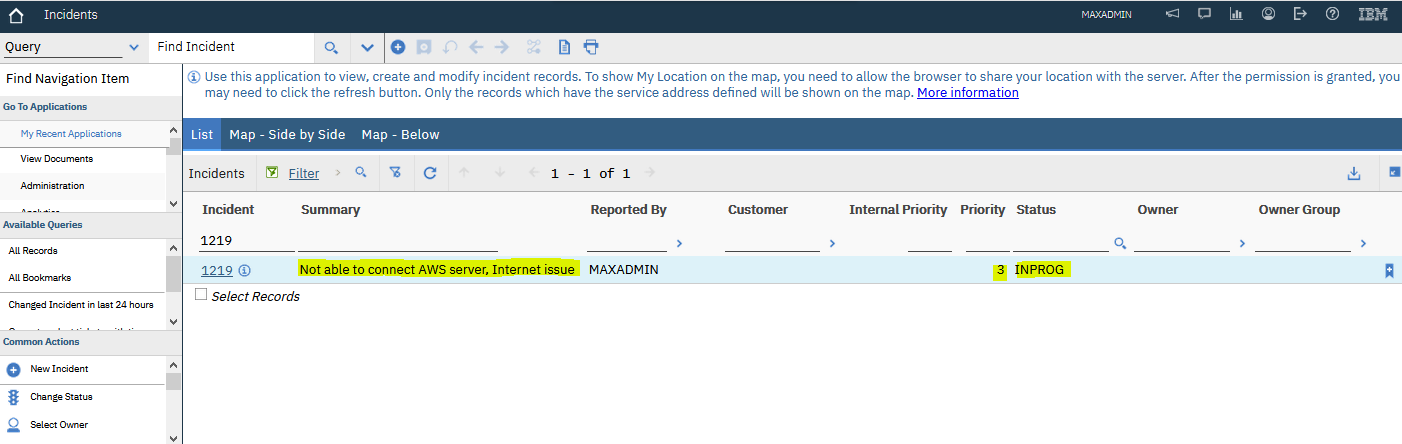
**ICD**

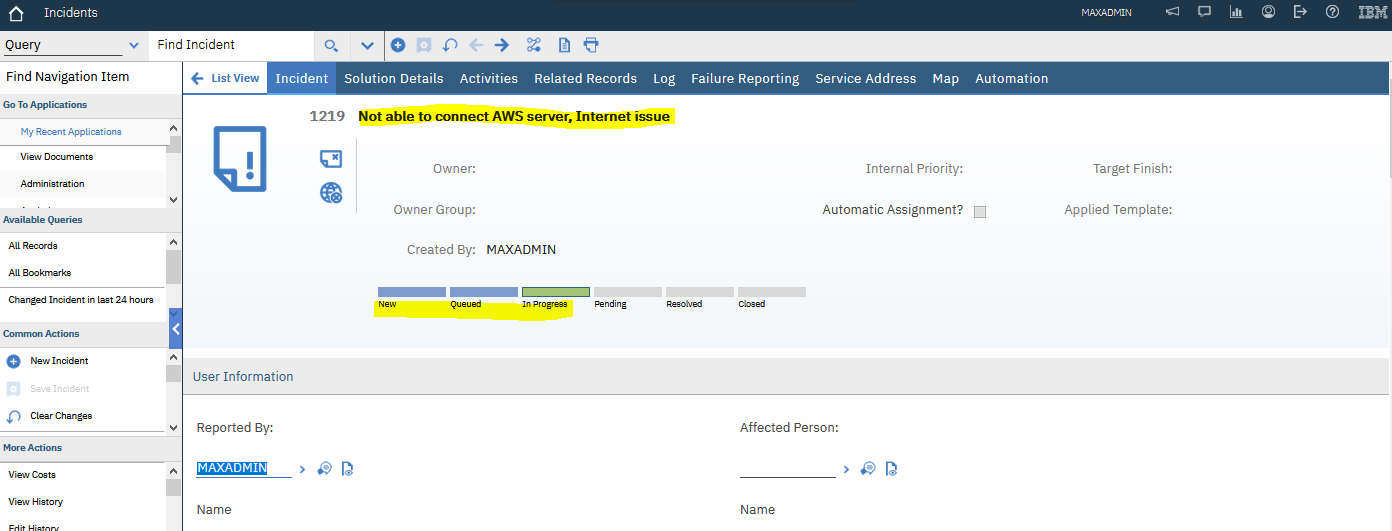
**JIRA**

1. Update the JIRA defect summary, webhook will be triggered and will be updated in ICD

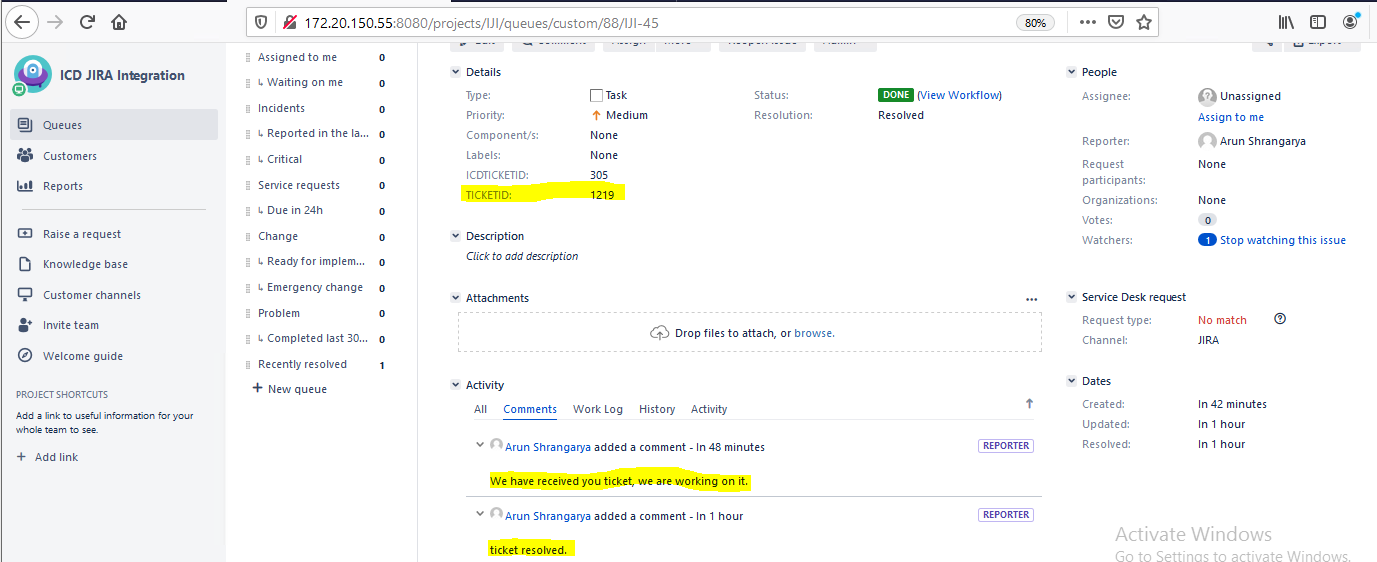
**JIRA**

**ICD**

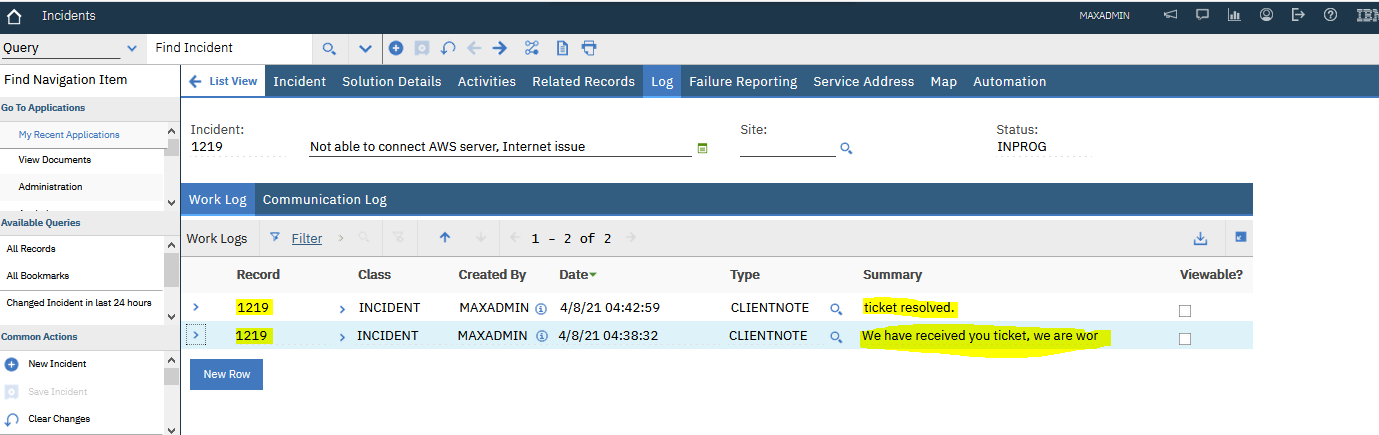




**JIRA**



**ICD**



# Incident creation using different user

