

Automated Network Request Management in ServiceNow

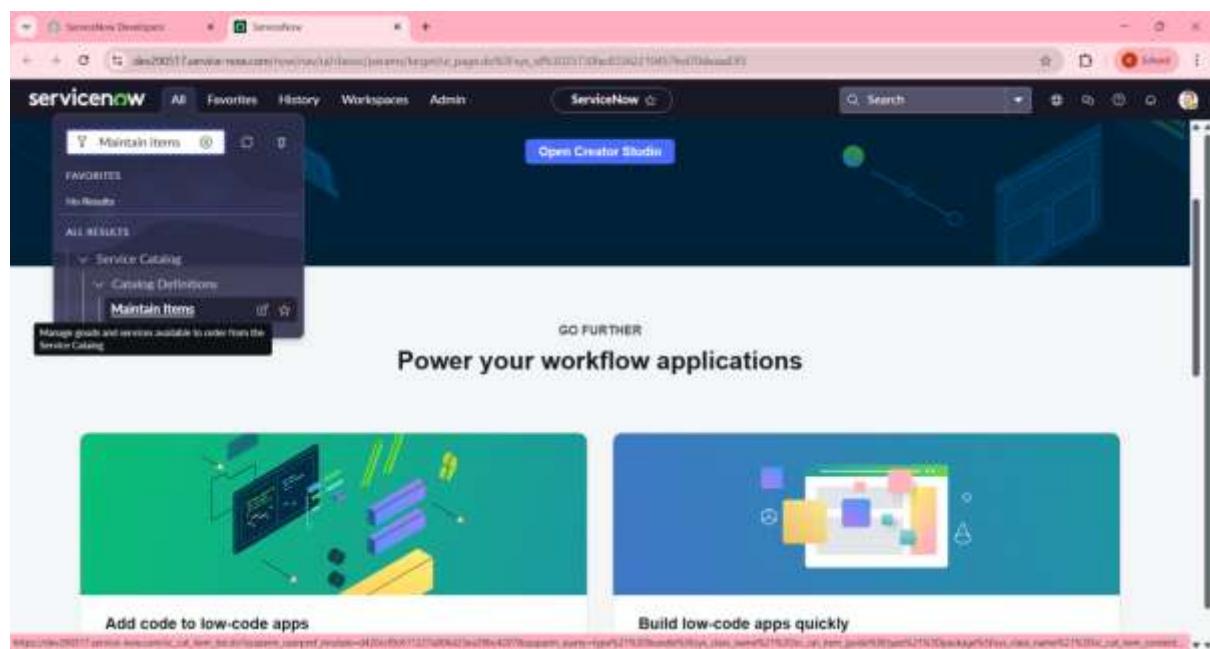
INTRODUCTION:

This project provides an automated solution in **ServiceNow** to manage network-related service requests. Through a self-service portal, users can easily submit requests, which are then validated, approved, and routed for fulfillment. Automated workflows handle approvals, notifications, and task assignments, while optional integrations with network tools reduce manual effort. The system also offers real-time updates and reporting to improve efficiency, transparency, and SLA tracking.

Process 1: Creation of Service Catalog – "Network Request"

Step 1: Navigate to Service Catalog

1. Open the **Application Navigator** in ServiceNow.
2. Go to:
All → Service Catalog → Maintain Items



Step 2: Create New Catalog Item

1. Click on **New**.
2. Fill the following details:
 - **Name:** Network Request
 - **Catalog:** Service Catalog
 - **Category:** Network and connectivity
 - **Short Description:** Network Request Management
3. Click on **Save**.

The screenshot shows the ServiceNow interface for creating a new catalog item. The title bar says "Catalog Item - Network Request". The main area has a header with "Name: Network Request", "Catalog: Service Catalog", "Category: Networks and Connectivity", and "Short description: Network request Management". There are also "Active" and "Fulfillment automation level" fields. Below the form is a rich text editor for the full description.

Step 3: Configure Variables

1. Open the newly created **Network Request** catalog item.
2. Scroll down to the **Variables** related list → Click **New** for each variable.
3. Fill out the following for each variable:
 - **Type:** Single line text, Choice, Reference, etc.
 - **Order:** e.g., 100, 200, 300 (controls display order)
 - **Question:** Label shown on the form
 - **Name:** Technical name (used in scripts)

- **Tooltip:** Info shown on mouse hover
- **Example Text:** Placeholder help text
- **Mandatory / Read-Only:** As required
- **Auto-populate:** Use dot-walking for dependent values

The screenshot shows the ServiceNow Catalog Item configuration interface. At the top, there's a navigation bar with tabs like 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. Below that is a search bar and a toolbar with buttons for 'Copy', 'Update', and 'Delete'. The main area is titled 'Catalog Items'.

The configuration details for the variable 'Is this a New connection or Relocation?' are displayed. The 'Type' is set to 'Multiple Choice'. The 'Question' field contains the text 'Is this a New connection or Relocation?'. The 'Name' field is 'is_this_a_new_connection_or_relocation'. The 'Order' is set to 300. There are also fields for 'Active' (checked), 'Mandatory' (unchecked), 'Readonly' (unchecked), 'Hidden' (unchecked), and 'Unique' (unchecked). Below the main configuration, there are tabs for 'General', 'Annotation', 'Type Specifications', 'Default Value', 'Auto-populate', 'Inheritance', and 'Availability'. The 'General' tab is selected, showing the question and name fields again, along with other optional fields like 'Conventional Label' and 'Tooltip'.

Step 4: Variable Types Configuration

Type	Question	Order
Container Start	Service Details	200
Multiple Choice	Is this a new network connection or a relocation?	300
Single Line Text	If this is a relocation, Please provide ...	310
Single Line Text	If this is a relocation, Please provide ...	320
Container Start	Location & Devices Type	400
Single Line Text	Please provide address here	410
Select Box	Type of devices	420
Single Line Text	Provide device details	430
Container Start	Additional Information	500

Single Line Text

If any, Please write here

510

The screenshot shows the ServiceNow interface for a catalog item named "Network Request". The top navigation bar includes "Catalog Item - Network Request", "Search", and "Edit in Catalog Builder". Below the header, there are tabs for "Variables (10)", "Variable Sets", "Catalog UI Policies", etc. A table titled "Assigned Topics" lists 10 variables. The columns are: Type, Readonly, Question, Name, Order, and Created. The variables are:

Type	Readonly	Question	Name	Order	Created
Container Start	false	Service Details	service_details	200	2025-07-25 11:17:03
Multiple Choice	false	Is this a new network connection or a re...	is_this_a_new_network_connection_or_a_re...	300	2025-07-25 11:22:08
Single Line Text	false	Is this a relocation, please provide y...	is_this_a_relocation,_please_provide_y...	310	2025-07-25 11:24:00
Single Line Text	false	If this is a relocation, Please provide ...	if_this_is_a_relocation,_please_provide_y...	320	2025-07-27 00:21:17
Container Start	false	Location & Device Type	location_device_type	400	2025-07-27 00:22:27
Single Line Text	false	Please provide address here:	please_provide_address_here	410	2025-07-27 00:23:56
Select Box	false	Type of devices	type_of_devices	420	2025-07-27 00:24:48
Single Line Text	false	Provide device details	provide_device_details	430	2025-07-27 00:25:31
Container Start	false	Additional Information	additional_information	500	2025-07-27 00:26:09
Single Line Text	false	If any, Please write here	if_any,_please_write_here	510	2025-07-27 00:28:07

Step 5: Configure Variable Set – Requester Information

5.1 Create Variable Set

1. Navigate to **Variable Sets** under Service Catalog.

The screenshot shows the ServiceNow interface for the "Variables" page. The top navigation bar includes "Variables (10)", "Variable Sets (1)", "Catalog UI Policies", etc. Below the header, there are tabs for "Variables (10)", "Variable Sets (1)", "Catalog UI Policies", etc. A table titled "Assigned Topics" lists 1 variable. The columns are: Type, Readonly, Question, Name, Order, and Created. The variable is:

Type	Readonly	Question	Name	Order	Created
Variable Set	false	Requester information	requester_information	100	2025-07-27 00:28:07

2. Click on **New**.

3. Fill the following details:

- o **Title:** Requester information
- o **Internal Name:** requester_information (auto-filled)
- o **Order:** 100
- o **Type:** Single Row
- o **Layout:** 2 Columns Wide, one side, then the other

- Check the box: **Display title**

4. Click **Submit or Update**

The screenshot shows the 'Variable Set - Requester information' configuration page. The 'Title' is set to 'Requester information'. The 'Internal name' is 'requester_information'. The 'Order' is 100. The 'Type' is 'Single Row'. The 'Display title' checkbox is checked. The 'Layout' is set to '2 Columns Wide, one side, then the other'. The 'Description' field is empty. At the bottom, there are tabs for 'Policies', 'Catalog Client Scripts', 'Included In {1}', and 'Catalog Data Lookup Definitions'. There is also a search bar and an 'Actions' button.

Step 5.2: Add Variables to the Variable Set "Requester Information"

After creating the variable set, now it's time to add the variables one by one.

1. Opened on behalf of

- Type: **Reference**
- Reference to: **User [sys_user]**
- Name: `opened_on_behalf_of`
- Order: 100
- This allows the requester to select a user they are raising the request for.

The screenshot shows the ServiceNow 'Variable - New Record' page. At the top, there are tabs for 'Variable' and 'New record'. Below the tabs, the 'Application' is set to 'Global'. The 'Type' is 'Single Line Text', 'Order' is 100, and it is 'Varibased' on 'Requester information'. On the right, there are checkboxes for 'Active' (checked), 'Mandatory' (unchecked), 'Read only' (unchecked), and 'Hidden' (unchecked). Below these settings, there is a tabbed section with 'Question' selected. The 'Question' field contains 'email_id'. The 'Name' field also contains 'email_id'. There are also fields for 'Conversational label', 'Tooltip', and 'Example Text', all of which contain 'email_id'. A 'Submit' button is at the bottom left.

2. Email ID

- Type: **Single Line Text**
- Name: **email_id**
- Order: 200
- This will be auto-filled based on the user selected in "Opened on behalf of".
- You can use a script or dot-walking to populate the email field.

3. User Name

- Type: **Single Line Text**
- Name: **user_name**
- Order: 300
- This will also be auto-populated based on the user selected.
- Fetch the full name from the User table.

4. Phone Number

- Type: **Single Line Text**
- Name: **phone_number**

- Order: 400
- Same as above, it can be fetched using dot-walking or client script.

5. Proof of Document

- Type: **Attachment**
- Name: **proof_of_document**
- Order: 500
- This allows users to upload a file (such as proof or ID documents).

The screenshot shows the ServiceNow Catalog Variable Sets interface. At the top, there's a navigation bar with tabs like 'All', 'Favorites', 'History', 'Workspaces', and a search bar. Below the navigation is a toolbar with 'Update' and 'Delete' buttons. The main area displays a table of variables. The table has columns for 'Name', 'Type', 'Question', and 'Order'. There are five rows in the table:

Name	Type	Question	Order
email_id	Single Line Text	Email ID	300
opened_on_behalf_of	Reference	Opened on behalf of	100
phone_number	Single Line Text	Phone Number	200
user_name	Single Line Text	User Name	200
proof_of_document	Attachment	Proof of Document	300

When a user is selected in the **Opened on behalf of** field, we want to automatically populate:

- Email ID
- User Name
- Phone Number

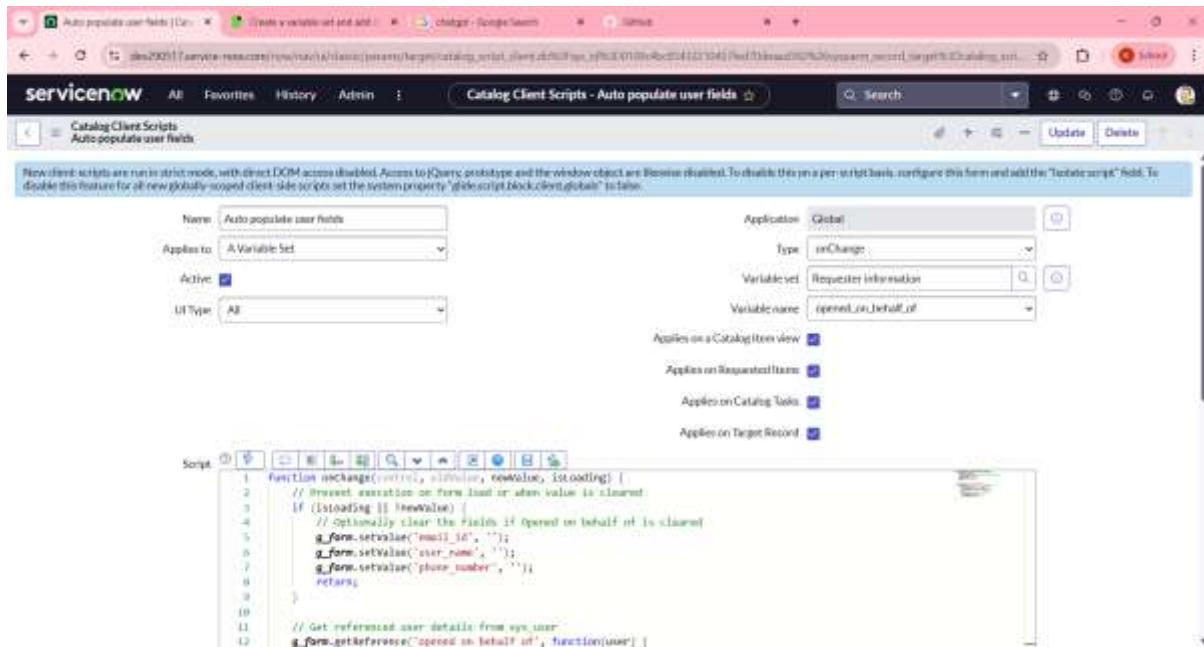
Steps to Auto-populate Fields

1. Open the Variable Set

- Navigate to: **Service Catalog > Catalog Variable Sets**
- Open your variable set: **Requester Information**

2.Create a Catalog Client Script

- Navigate to: **Service Catalog > Catalog Client Scripts**
- Click **New**
- Fill in details:
 - **Name:** Auto Populate User Info
 - **Applies to:** Catalog Item
 - **Variable Set:** Select *Requester Information*
 - **UI Type:** All
 - **Type:** onChange



3.Configure the Script Fields

- **Variable name:** opened_on_behalf_of
- **Script:**

The screenshot shows the configuration dialog for the 'opened_on_behalf_of' variable. The **Script** field contains the same onChange function as shown in the previous screenshot.

```
1 function onChange(currentValue, newValue, isLoading) {
2     // Prevent execution on form load or when value is cleared
3     if (!isLoading || !newValue) {
4         // Optionally clear the fields if opened on behalf of is cleared
5         g_form.setValues('email_id', '');
6         g_form.setValues('user_name', '');
7         g_form.setValues('phone_number', '');
8         return;
9     }
10
11     // Get referenced user details from sys_user
12     g_form.getReference('opened_on_behalf_of', function(user) {
13
14     // Set values
15     g_form.setValues('email_id', user.email || '');
16     g_form.setValues('user_name', user.name || '');
17     g_form.setValues('phone_number', user.mobile_phone || '') // updated line
18
19     // Fallback in case user couldn't be retrieved
20     if (!user) {
21         g_form.setValues('email_id', '');
22         g_form.setValues('user_name', '');
23         g_form.setValues('phone_number', '');
24     }
25 })
```

Step 6: Catalog UI Policy Configuration

Goal: Show " Provide device details here " field when **Types of Devices = Others**.

1. Navigate to the **Network Request** catalog item.
2. In the related list, go to **Catalog UI Policies** → Click **New**.
3. Fill in:
 - **Applies to:** Catalog Item
 - **Catalog Item:** Network Request
 - **Condition:** Types of devices is Others
4. Click **Save**.

5. In the related list, click **New** under **UI Policy Actions**.
6. Set:
 - **Catalog Item:** Network Request
 - **Variable name:** Provide device details here
 - **Visible:** True
7. Click **Update** to save policy.
8. **Test the form** to ensure the field appears based on selection.

Catalog UI Policy
New record

Catalog UI policies are similar to standard UI policies. Catalog UI policies dynamically change variables that are part of a catalog item or change how variable sets are handled. Policies can also be applied when the variables are present in a Requested Item or Catalog Task form. [More Info](#)

Applies to	Application
A Catalog Item	Global
* Catalog item	Active
Network Request	<input checked="" type="checkbox"/>
* Short description	
Display field when device is Others	

When to Apply **Script**

Catalog UI policy actions are applied only if all the following conditions are met:

1. The catalog UI policy is **Active**
2. The items in the **Conditions** field evaluate to true
3. The field specified in the catalog UI policy is present on the specified catalog item

Catalog Conditions

Add Filter Condition Add "OR" Clause

type_of_devices is Others

Applies on a Catalog Item view

33°C Partly sunny

Search

Process 2: Creation of Table and Fields in ServiceNow

>Network Database Table

Step 1: Create a New Table

1. Navigate to the Application Navigator.
2. Type: Tables under the **System Definition** module.
3. Click on **Tables**.
4. On the top-right corner, click on **New** to create a new table.

5. Fill in the table details:

- **Label:** *Network Database Table*
- **Name:** Automatically generated (or customize if needed).
- Keep **Auto-generate schema** checked.

6. Click **Submit** to create the table.

The screenshot shows the ServiceNow 'Table - New Record' page. At the top, there are fields for 'Label' (set to 'Network Database Table') and 'Name' (set to 'network_database_table'). Below these are sections for 'Application' (set to 'Global'), 'Create module' (checkbox checked), 'Create reusable module' (checkbox checked), and 'Add module to menu' (dropdown set to 'Create new...'). A note at the top says 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.' The main area shows a table with columns for 'Column label', 'Type', 'Reference', 'Maxlength', 'Default value', and 'Display'. A button at the bottom left says 'Insert a new row...'.

Step 2: Add custom fields

These fields are **custom fields** that you will manually add in the Table Columns section of your custom table.

1. Name: u_request_number

- **Label:** Request Number
- **Type:** String
- **Reference:** —
- **Explanation:** A unique identifier for the request. Can be filled manually or auto-generated using a Business Rule.

2. Name: u_assignment_group

- **Label:** Assignment Group
- **Type:** Reference
- **Reference:** Group (Group table)
- **Explanation:** Defines the team or group responsible for fulfilling the request.

3. Name: u_customer_document

- **Label:** Customer Document
- **Type:** String
- **Reference:** —
- **Explanation:** Stores a document reference or identifier related to the customer, such as an ID proof or contract reference

4. Name: u_assigned_to

- **Label:** Assigned To
- **Type:** Reference
- **Reference:** User(User table)
- **Explanation:** The specific user assigned to handle the request.

5. Name: u_device_details

- **Label:** Device Details
- **Type:** String
- **Reference:** —
- **Explanation:** Captures technical details or specifications of the device involved in the request.

6. Name: u_date_of_enquiry

- **Label:** Date of Enquiry
- **Type:** Date
- **Reference:** —

- **Explanation:** The date when the enquiry was received from the customer.

7. Name: u_customer_address

- **Label:** Customer Address
- **Type:** String
- **Reference:** —
- **Explanation:** The physical or mailing address of the customer.

8. Name: u_approval_state

- **Label:** Work Status
- **Type:** String
- **Reference:** —
- **Explanation:** Indicates the current approval or work status of the request.

9. Name: u_requested_for

- **Label:** Requested For
- **Type:** String (*Normally this should be a Reference to sys_user, but in your screenshot it's String*)
- **Reference:** — (*unless you change it to a Reference type*)
- **Explanation:** Specifies the end-user for whom the request is being made.

The screenshot shows the ServiceNow interface for configuring a Network Database Table. The top navigation bar includes 'ServiceNow Developers', 'Network Database Table | Table', 'Properties and configurations', and a URL. Below the header, there's a breadcrumb trail: 'servicenow' > 'Tables' > 'Network Database Table'. The main content area is titled 'Table - Network Database Table' and contains a table with columns: Column, Type, Reference, Max length, Default value, and Display. The table lists various fields like 'Customer Address', 'Customer', 'Case', etc., with their respective data types and configurations. One row, 'Number Maintenance', is highlighted in blue.

Column	Type	Reference	Max length	Default value	Display
Customer Address	String	general	100		False
Customer	String	general	40		False
Case	System Case Number	general	30	[@caseNumberForSalePhone();]	False
Assignment Group	Reference	general	32		False
Owner	Case Owner	general	40		False
Requester	String	general	40		False
sys_id	System ID	general	32		False
Created By	String	general	100		False
Number Maintenance	String	general	40	1003	False
Customer Address	String	general	40		False
Analyst Id	Reference	User	32		False
Unknown	String	general	40		False
Master	Image	general	40		False
Work Group	String	general	40		False
Work Status	String	general	40	open	False
Customer Document	String	general	40		False
Request Number	String	general	40		False
Case Details	String	general	40		False
Deadline Number	String	general	40	[@caseNumberForSalePhone();]	False
Date of Creation	Date	general	40	[@dateStringForUser();]	False

To Autopopulate Database Number

Using Number Maintenance

ServiceNow has a built-in feature called **Number Maintenance** to manage auto-number sequences for any table.

1. Navigate to:
System Definition > Number Maintenance.
2. Click **New**.
3. Fill in details:
 - o **Table** → select your Network Database Table.
 - o **Prefix** → NET.
 - o **Current Value** → 1003 (or any starting number you want).
 - o **Number of Digits** → 7.
4. Save.

The screenshot shows a ServiceNow web interface for creating a new record in the 'Network DataBase Table'. The URL in the address bar is `https://servicenow:8080/nmw/u/classic/jarvis/network_database_table.htm?sysparm_name=Create`. The page title is 'Network DataBase Table - Create NET0001029'. The form contains the following fields:

- Database Number: NET0001029
- Work Status: New
- Assignment Group: (dropdown menu)
- Assigned to: (dropdown menu)
- Device Details: (dropdown menu)
- Request Number: (text input)
- Created: (text input)
- Request For: (text input)
- Date of Enquiry: (text input)
- Customer Address: (text input)
- Special Instructions: (text input)

A 'Submit' button is located at the bottom left of the form area.

Network Task Table

Step 1: Create the Child Table (Network Task Table)

1. Navigate to:
System Definition > Tables
2. Click **New**.
3. Fill in details:
 - o **Label** → Network Task Table
 - o **Name** → auto-generated (u_network_task_table)
 - o **Extends Table** → select **Network Database Table** (u_network_database_table)

This is the important part → by choosing **Extends Table**, your Network Task Table will automatically inherit all fields from the parent.

4. Save the record.

The screenshot shows the ServiceNow interface for managing tables. The title bar says 'Table - Network Task Table'. The top navigation includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and 'Table - Network Task Table'. Below the title bar, there's a message: 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes.' A 'More Info' link is present. The main area shows the table configuration with fields for 'Label' (Network Task Table), 'Name' (a.network_task_table), and 'Extends table' (Network DataBase Table). Below this, the 'Columns' tab is selected, showing a list of columns with details like type (String, Reference), length, default value, and display status. Fields listed include Gateway Address, Created By, Work Notes, Class, Description, Assignment Group, Created, and Request For.

Column label	Type	Reference	Maxlength	Default value	Display
Gateway Address	String	(empty)	500		false
Created By	String	(empty)	40		false
Work Notes	String	(empty)	40		false
Class	System Class Name	(empty)	80	!script(current.getTableNamed()	false
Description	String	(empty)	40		false
Assignment Group	Reference	Group	32		false
Created	Date/Time	(empty)	40		false
Request For	String	(empty)	40		false

Step 2: Verify Inherited Fields

- Open the new table (Network Task Table).
- Go to **Columns** tab.
- You'll see:
 - Fields from parent (Database Number, Request Number, Request For, etc.)
 - Plus any new fields you add specifically for tasks (Task Number, Work Status, Assigned to, etc.).

The screenshot shows the ServiceNow interface for configuring the Network Task Table. The top navigation bar includes tabs like 'System Definition', 'Network Task Table', 'Create NTT000', 'Payments', 'Service', 'Log In', 'Search', 'Comments', 'Auto-populator', and 'New Job'. The main area is titled 'Table - Network Task Table' and shows a list of fields with their properties. Fields include 'Customer Address', 'Customer ID', 'Work Notes', 'User', 'Description', 'Assignment Group', 'Created', 'Request Ref', 'Task ID', 'Database Number', 'Special Instructions', 'Customer Address', 'Assigned To', 'Updated By', 'Updated On', 'Work Notes', 'Work Status', 'Approved Status', 'Customer Document', and 'Request Number'. The 'Display' column indicates which fields are visible. A 'Default value' column contains some custom JavaScript code, such as 'jso\$.getCustomer.getSalesPerson()'. At the bottom, there are buttons for 'Delete', 'Update', and 'Delete All Records'.

Field	Type	Reference	Max length	Default value	Display
Customer Address	String	(empty)	255		True
Customer ID	String	(empty)	40		True
Work Notes	String	(empty)	40		True
User	SystemUser	(empty)	30	jso\$.getCustomer.getSalesPerson()	False
Description	String	(empty)	40		True
Assignment Group	Reference	(empty)	10		True
Created	DateTime	(empty)	40		True
Request Ref	String	(empty)	40		False
Task ID	String	(empty)	10		True
Database Number	String	(empty)	255		True
Special Instructions	String	(empty)	40		True
Customer Address	String	(empty)	40		True
Assigned To	Reference	(empty)	10		True
Updated By	String	(empty)	40		True
Updated On	String	(empty)	40		True
Work Notes	String	(empty)	40		True
Work Status	String	(empty)	40	Open	True
Approved Status	String	(empty)	40		True
Customer Document	String	(empty)	40		True
Request Number	String	(empty)	40		True
Inherited from...					

Step 3: Configure Auto Numbering for Task Table

If you want separate auto numbering for **Network Tasks** (like NTT0001001):

1. Navigate to **System Definition > Number Maintenance**.
2. Click **New**.
3. Fill details:
 - o **Table** → Network Task Table
 - o **Prefix** → NTT
 - o **Current Value** → 1001
 - o **Number of Digits** → 7
4. Save.

Now each task will have a unique Task Number (NTT0001001, NTT0001002 ...).

Step 4: Adjust the Form Layout

1. Open a record in **Network Task Table**.
2. Right-click the header → **Configure > Form Layout**.
3. Add inherited fields (Database Number, Request Number, etc.) and new fields (Task Number, Work Notes, etc.).
4. Arrange as you like.

Process 3: Request Approvals Creation

The goal is to display **approval records** directly on the **Network Database table** form.

By creating a relationship between **Network Database Table** and **Approval (sysapproval_approver)**:

- We can see which approvals are associated with each record.
- We avoid searching in a separate table.
- The refineQuery ensures only relevant approvals (based on source table and document ID) are shown.

Steps to Create the Related List with Script

1. Navigate to Relationships

1. Go to **System Definition → Relationships**.
2. Click **New**.

2. Fill in the Relationship Details

- **Name** → Request Approvals

- **Applies to table** → Network Database Table [u_user_network_database]
- **Queries from table** → Approval [sysapproval_approver]
- **Active** → Checked.

3. Add the refineQuery Script

The script filters the approvals to only show records related to the current Network Database record.

```
(function refineQuery(current, parent) {
    current.addQuery('source_table', parent.getTableName());
    current.addQuery('document_id', parent.sys_id);
})(current, parent);
```

Script Explanation:

- **source_table** → Ensures only approvals linked to this specific table are fetched.
- **document_id** → Matches the approval record to the exact parent record.
- **state filter (commented out)** → Can exclude approvals not required.

4. Save and Verify

1. Click **Update**.
2. Open a **Network Database Table** record.
3. You should see the **Request Approvals** related list populated with the matching approval entries.

Steps to Add the Related List to the Form

1. Open any record from the **Network Database Table**.
2. Click the **context menu** (three dots in the top right of the form).
3. Navigate to **Configure > Related Lists**.
4. In the list of available related lists, select **Approval Request**.
5. Save the form configuration.

6. Refresh the record — you should now see the **Request Approvals** related list at the bottom of the form, displaying:

- **State**
 - **Approver**
 - **Comments**
 - **Approval for**
 - **Created**
-

Creation & Implementation of Flows, Actions in Flow Designer

Flow Designer in ServiceNow to automate the **Network Request** process.

The flow manages the entire lifecycle of a request — from capturing catalog variables, creating a record in the Network Database, sending notifications, requesting approvals, handling logic conditions, and updating records — all without manual intervention.

This ensures:

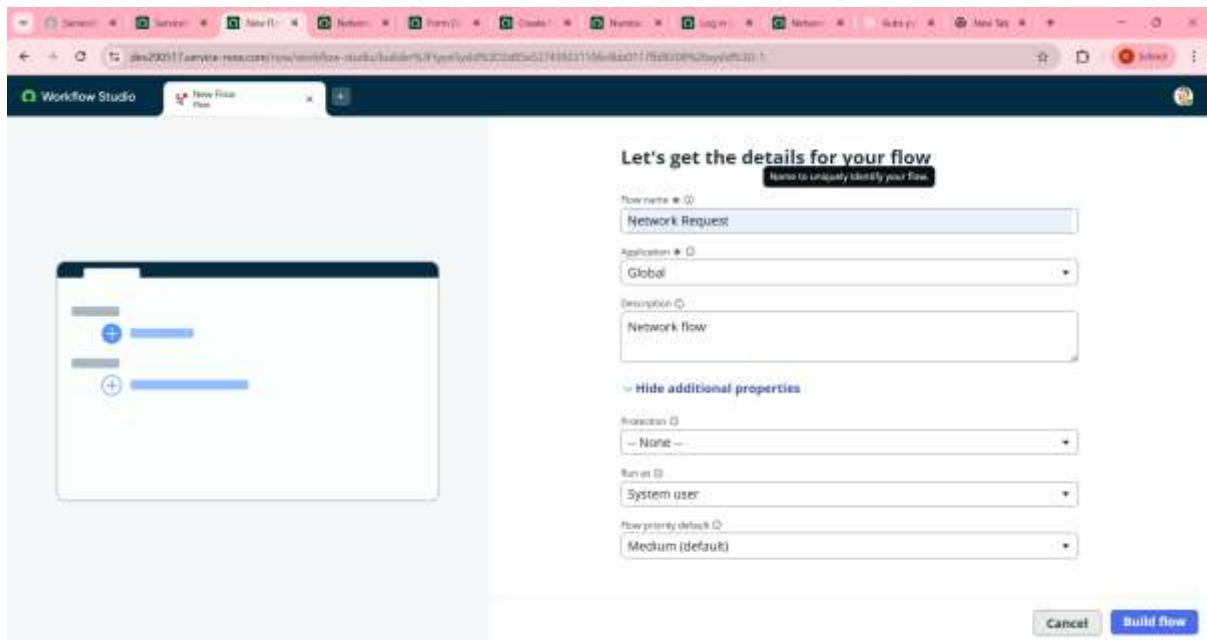
- Consistency in processing requests
- Faster execution
- Fewer manual errors
- Clear traceability of actions

Steps to Create the Flow

1. Creating the Flow

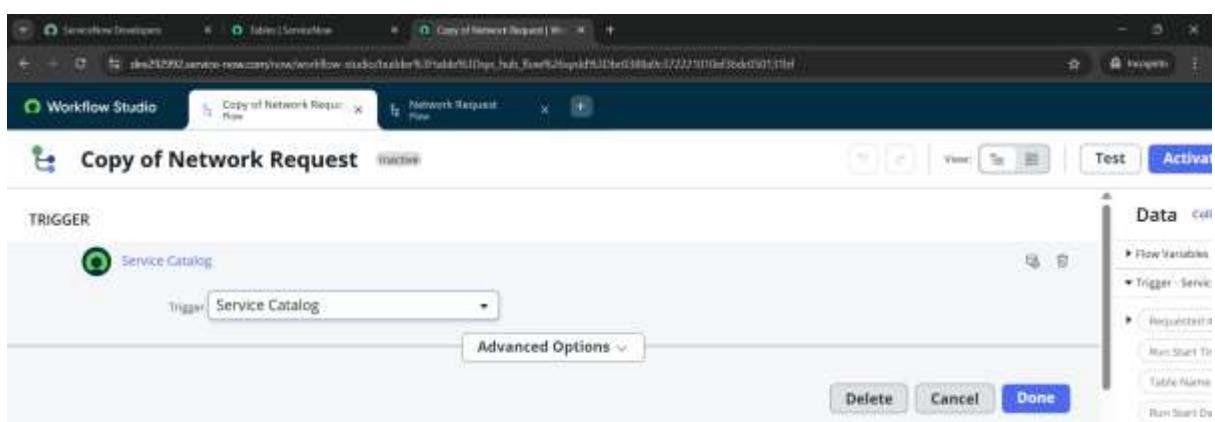
1. Navigate to **Flow Designer** home page.
2. Click **New** to create a new flow.
3. Enter:
 - **Flow Name:** Network Request
 - **Description:** (*e.g., Automates network request creation, approvals, and updates.*)

4. Click Build Flow.



2. Configuring the Trigger

1. Click the (+) icon to add a trigger.
2. Select:
 - o **Trigger Type:** Application → Service Catalog
3. Click Done.



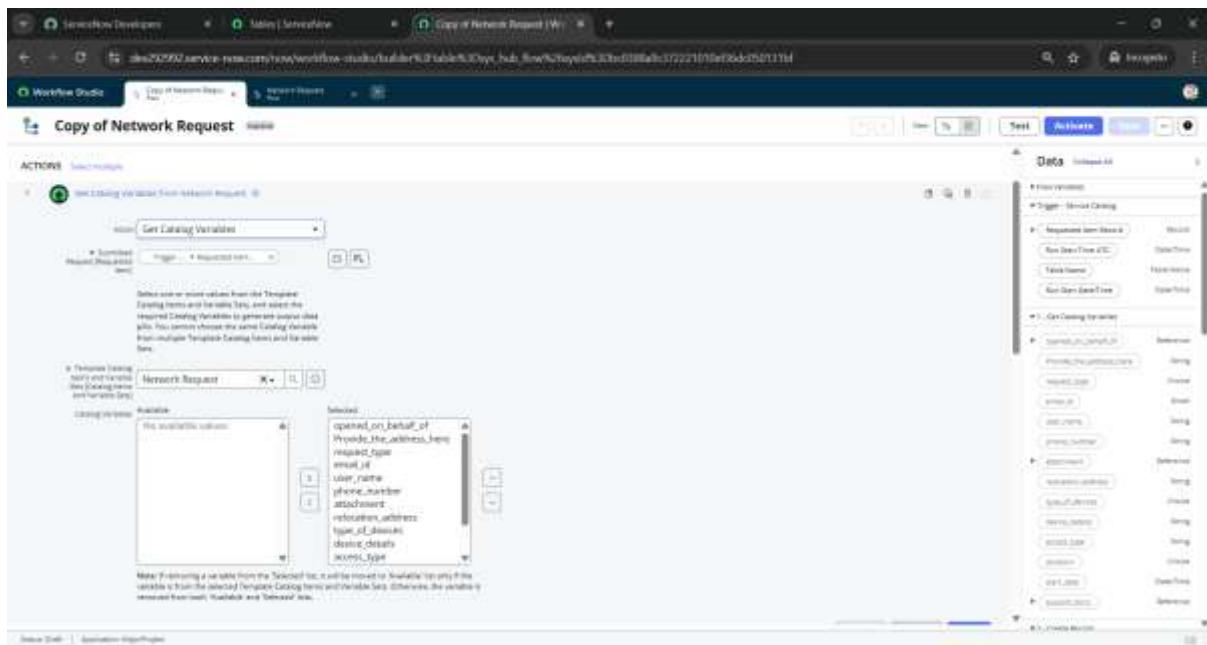
3. Adding Actions

A. Get Catalog Variables

1. Click **Actions**.
2. Search for **Get Catalog Variables**.

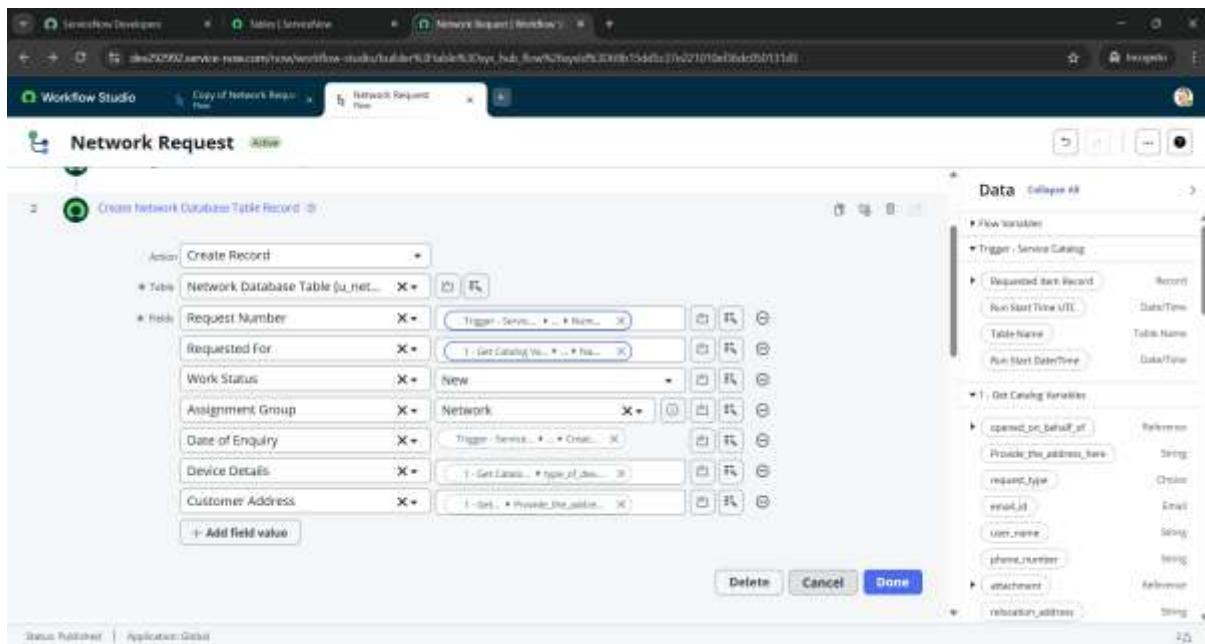
3. Select **Get Catalog Variables**.
4. Configure Action Inputs:
 - o Trigger → Service Catalog → Requested Item
5. In Template catalog items:
 - o **Select Table:** Network Request
 - o Move required variables to the **Selected** area.

6. Click **Done**.



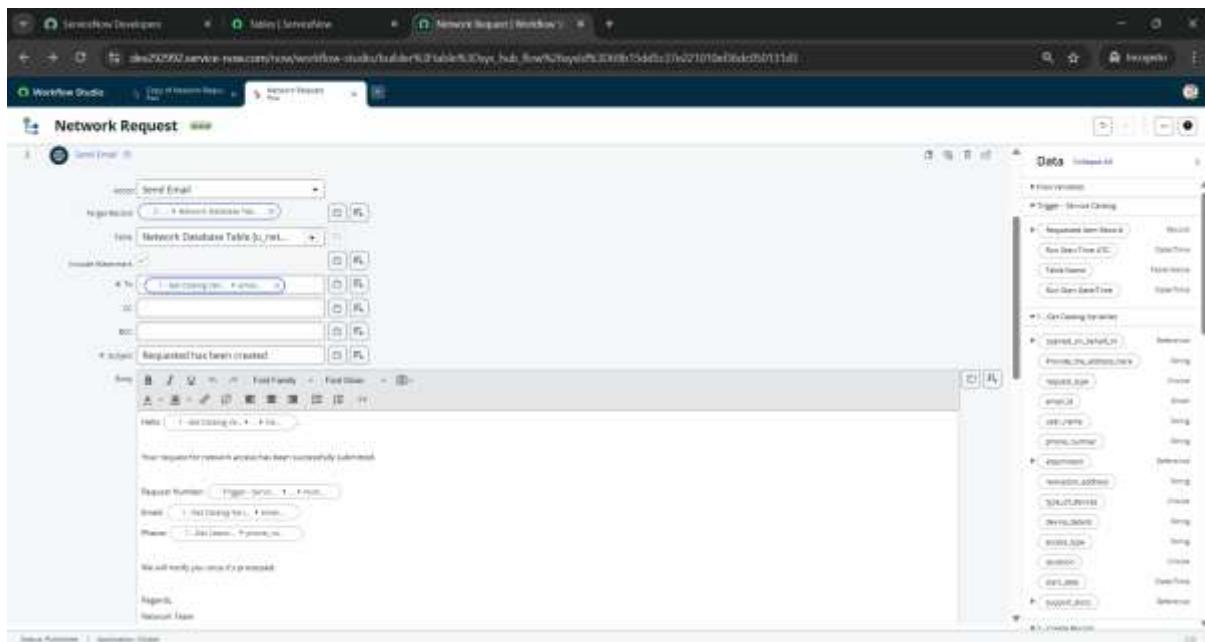
B. Create Record

1. Add a new action → **Create Record**.
2. Select **Table:** Network Database.
3. Click **Add Fields** and configure:
 - o Map catalog variables to the respective table fields as per your requirements .
4. Click **Done**.



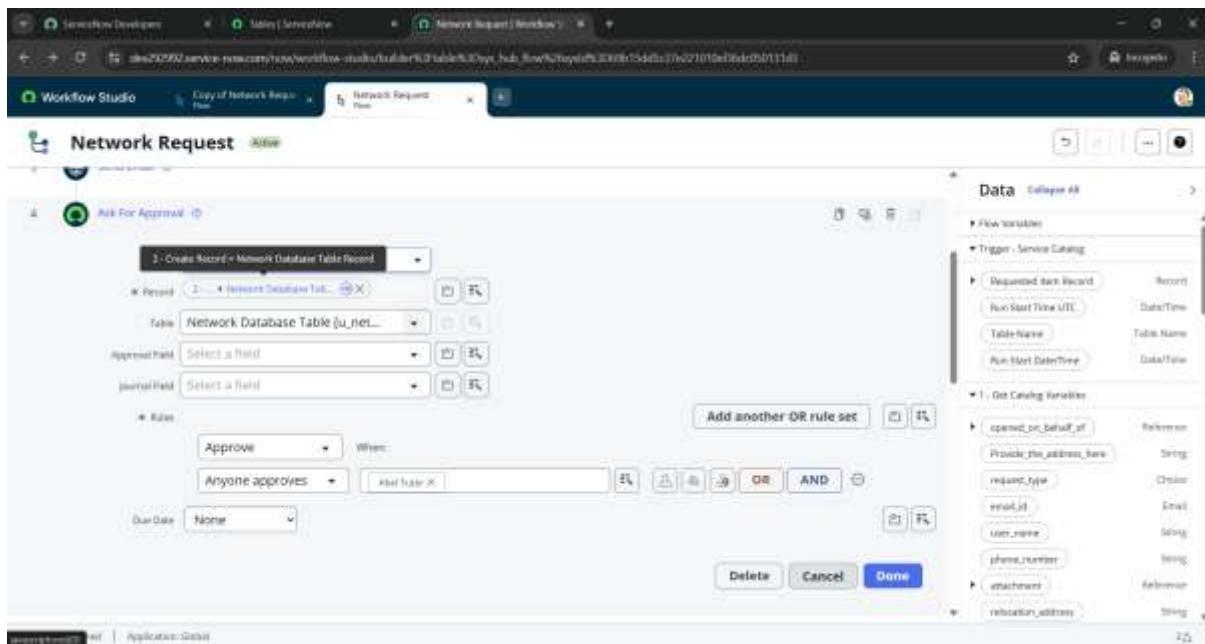
C. Send Email

1. Add a new action → **Send Email**.
2. **Target Record:** Select → **Create Record** → **Network Database Table** (auto-selected).
3. Configure:
 - **To / CC / BCC:** Static or dynamic recipients.
 - **Subject & Body:** Use variables and static text as shown in the design screenshot.
4. Click **Done**



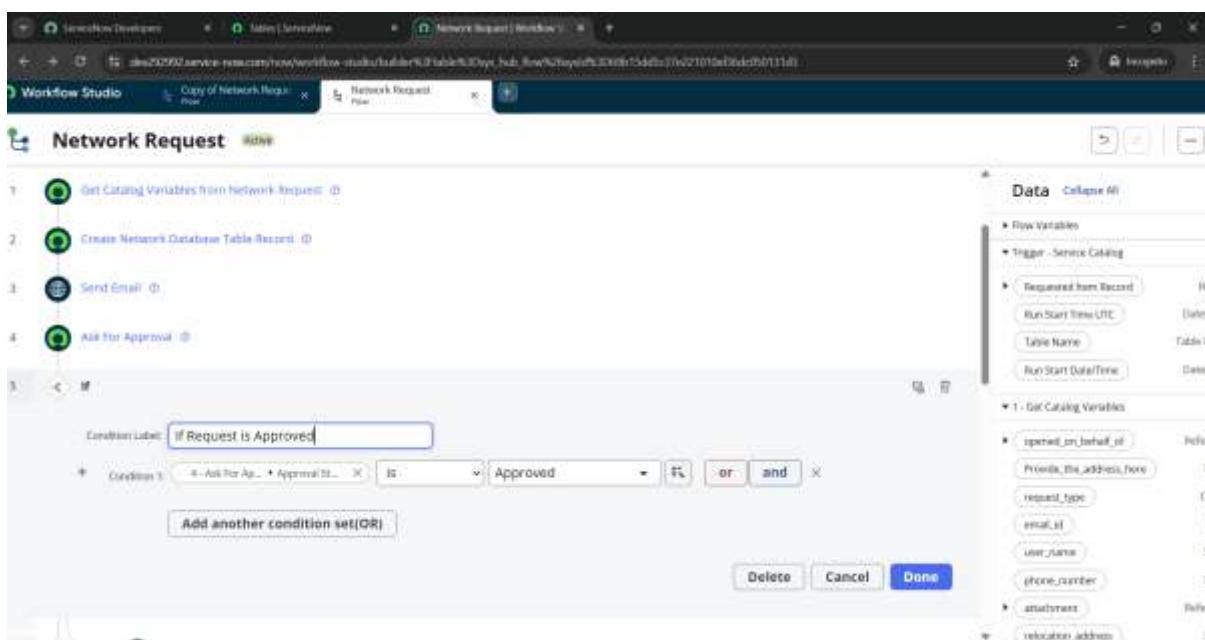
D. Ask for Approvals

1. Add a new action → **Ask for Approval**.
2. **Target Record:** Create Record → Network Database Table.
3. Configure:
 - **Approval Reason:** "Waiting for Approval".
 - **Approval Rules:** Approve, Reject, Approve/Reject.
 - **Approval Type:** Anyone approves, Everyone approves, etc. (static/dynamic assignment).
 - Here we chose abel tuter
4. Click **Done**



E. Flow Logic (If Condition)

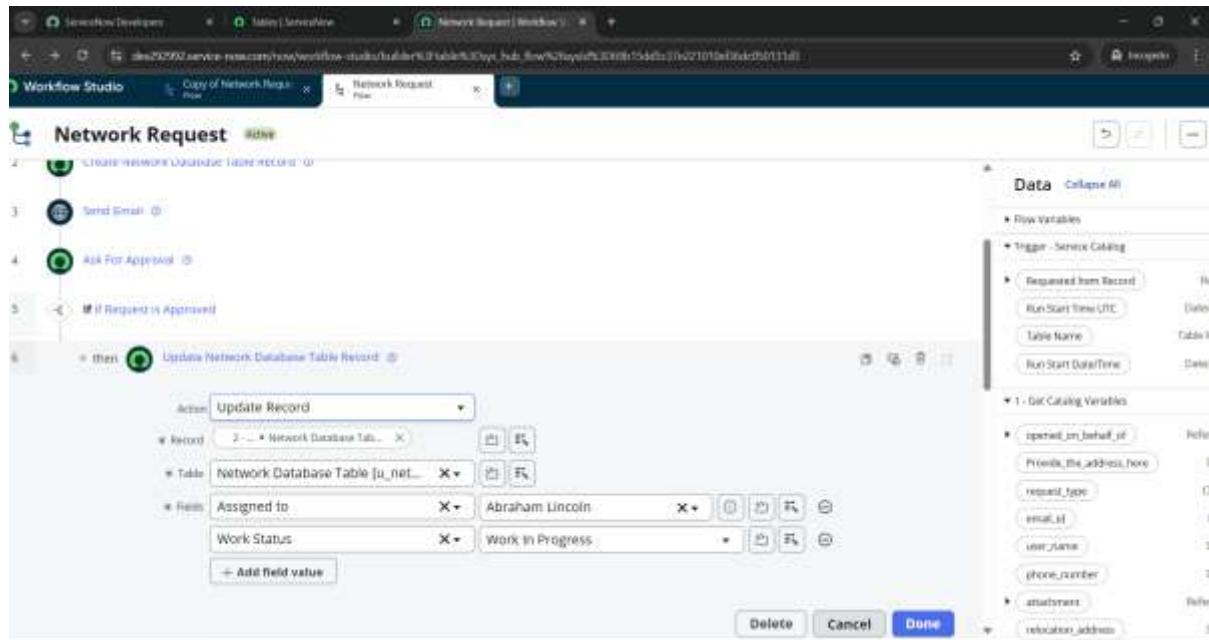
1. Add a new action → **Flow Logic** → **If Condition**.
2. Configure:
 - Condition: "Ask for approvals" state is Approved .
3. Click **Done**.



F. Update Record

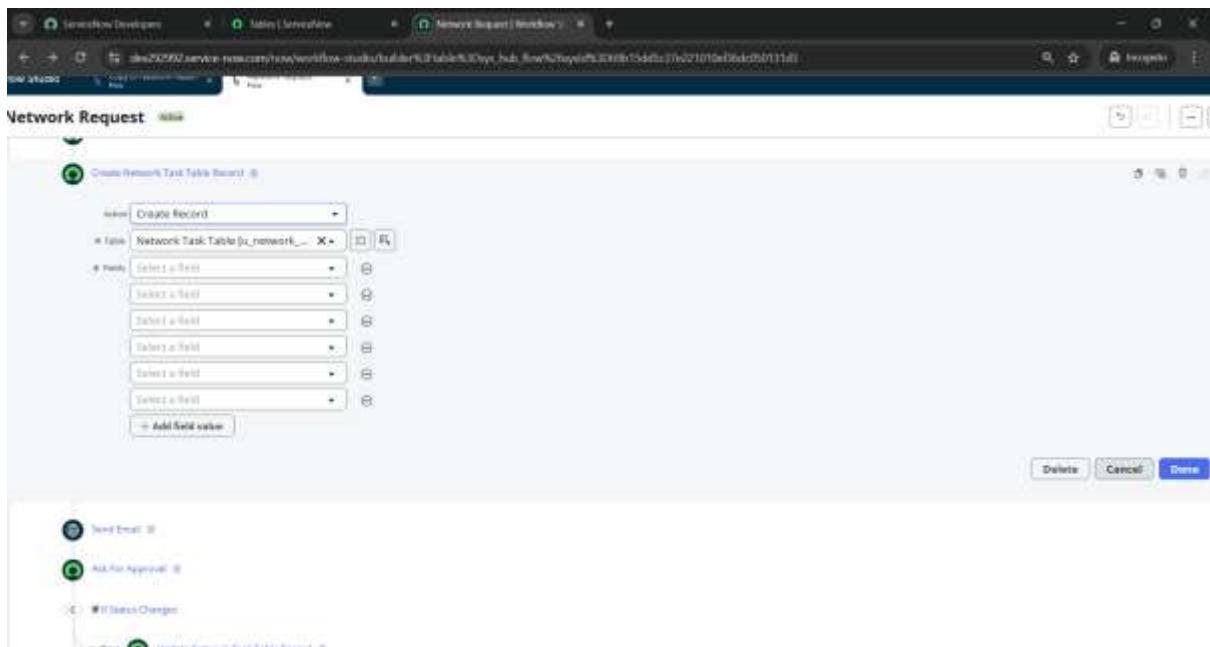
1. Add a new action → **Update Record**.

2. **Target Record:** Create Record → Network Database Table (auto-selected).
3. Configure required fields (like Assigned to -> Abraham Lincoln Work Status -> Work in Progress).
4. Click **Done**.



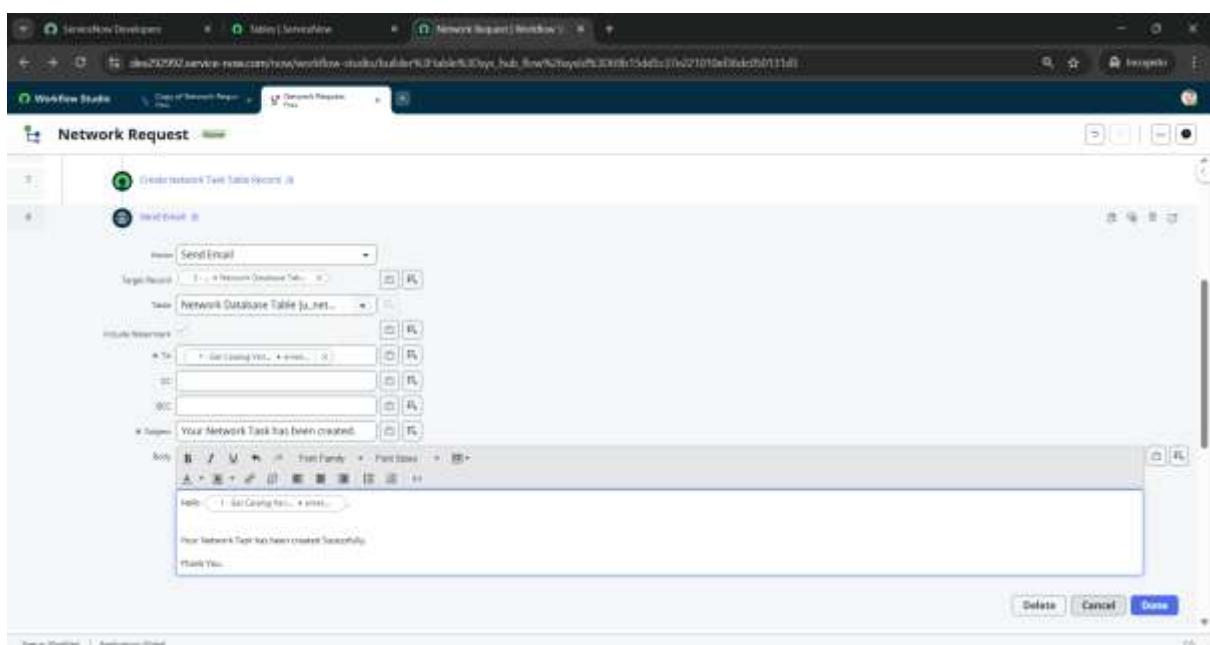
G: Create Network Task Table Record

1. Add a new action → **Create Record**.
2. Select **Table** → *Network Task Table [u_network_task]*.
3. Under **Fields**, map Service Catalog variables to the table fields:
 - o **Database Number** → Auto-populated (Number Maintenance / Business Rule).
 - o **Request Number** → Map from Catalog Variable (e.g., *Request Number*).
 - o **Requested For** → Map from Catalog Variable (Requested For).
 - o **Description** → Map from Catalog Variable (Description of request).
 - o **Priority** → Map from Catalog Variable (Priority).
 - o **Assignment Group** → Network Assignment Group (static or from variable).
 - o **Assigned To** → Leave blank initially (will be set later after approval).
4. Click **Done**.



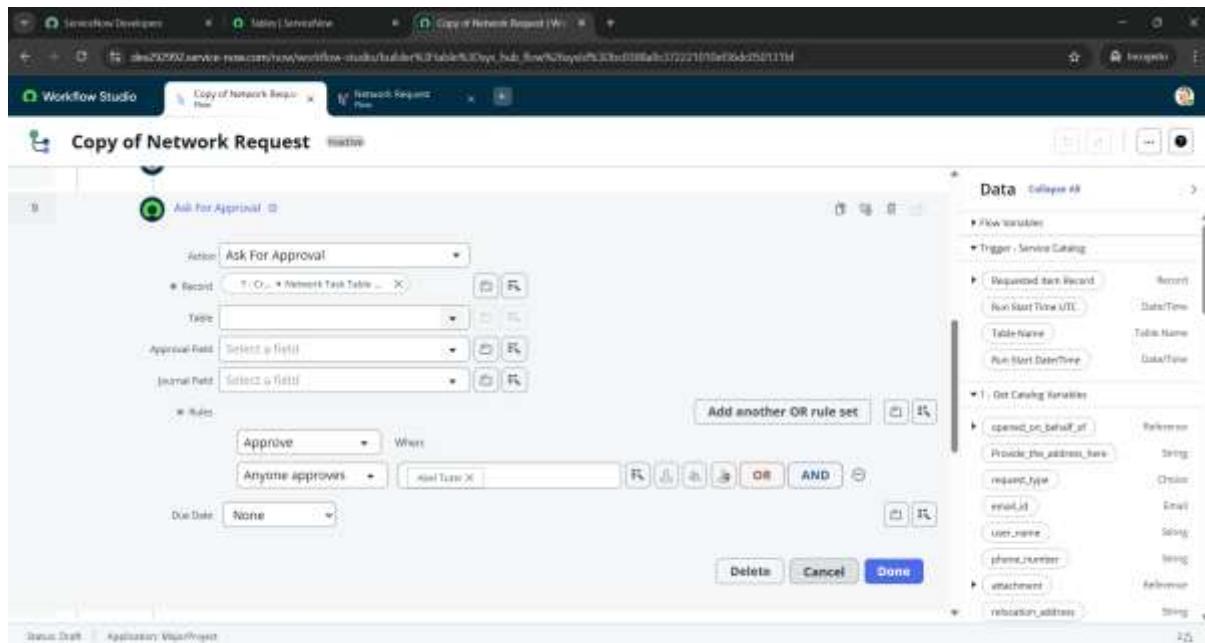
H. Send Email (Request Created)

1. Add a new action → **Send Email**.
2. Target Record → *Create Network Task Table Record*.
3. Configure:
 - o **To:** Requestor / Requested For.
 - o **Subject:** "Your Network Task has been created."
 - o **Body:** Include Task Number, Database Number, Request Number.
4. Click **Done**.



I. Ask for Approval

1. Add a new action → **Ask For Approval**.
2. Target Record → *Network Task Table Record*.
3. Configure:
 - o **Approval Reason:** "Waiting for Network Task approval".
 - o **Approval Rules:** Approve / Reject.
 - o **Approval Type:** Choose (e.g., *Anyone Approves*).
4. Click **Done**.



J. If Condition – Approval Status Changes

1. Add action → **If Condition**.
2. Condition → *Approval State is Approved*.
3. In the **Then branch**:

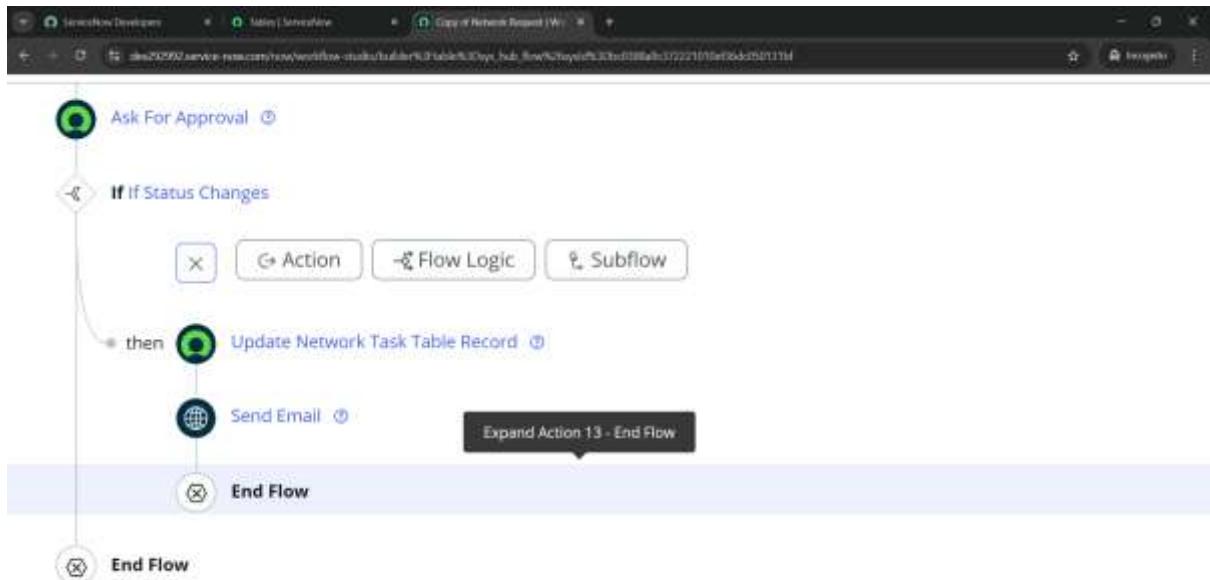
Update Record

- o Target Record → *Network Task Table Record*.
- o Update fields:
 - Assigned To → *Adam Ringle*.
 - Work Status → *Work in Progress*.
- o Click **Done**.

Send Email (Approved)

- Add action → Send Email.
- Notify requestor that the task is approved and in progress.

(same as above)



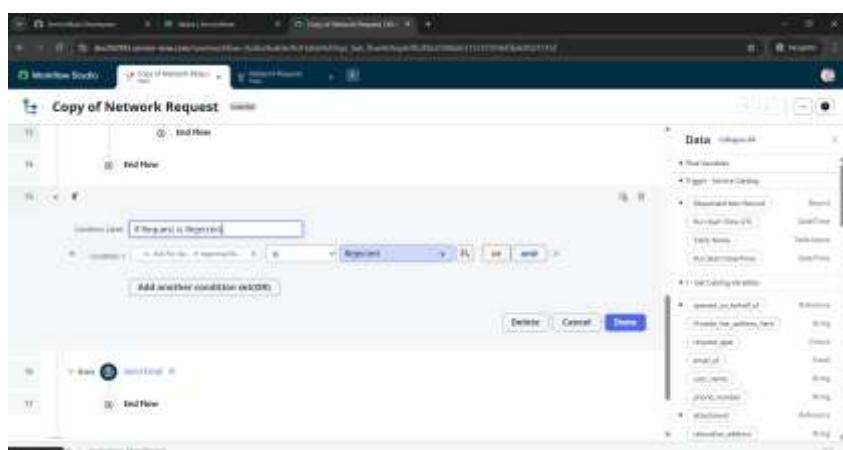
If Request is Rejected,

K. If Condition – Request Rejected

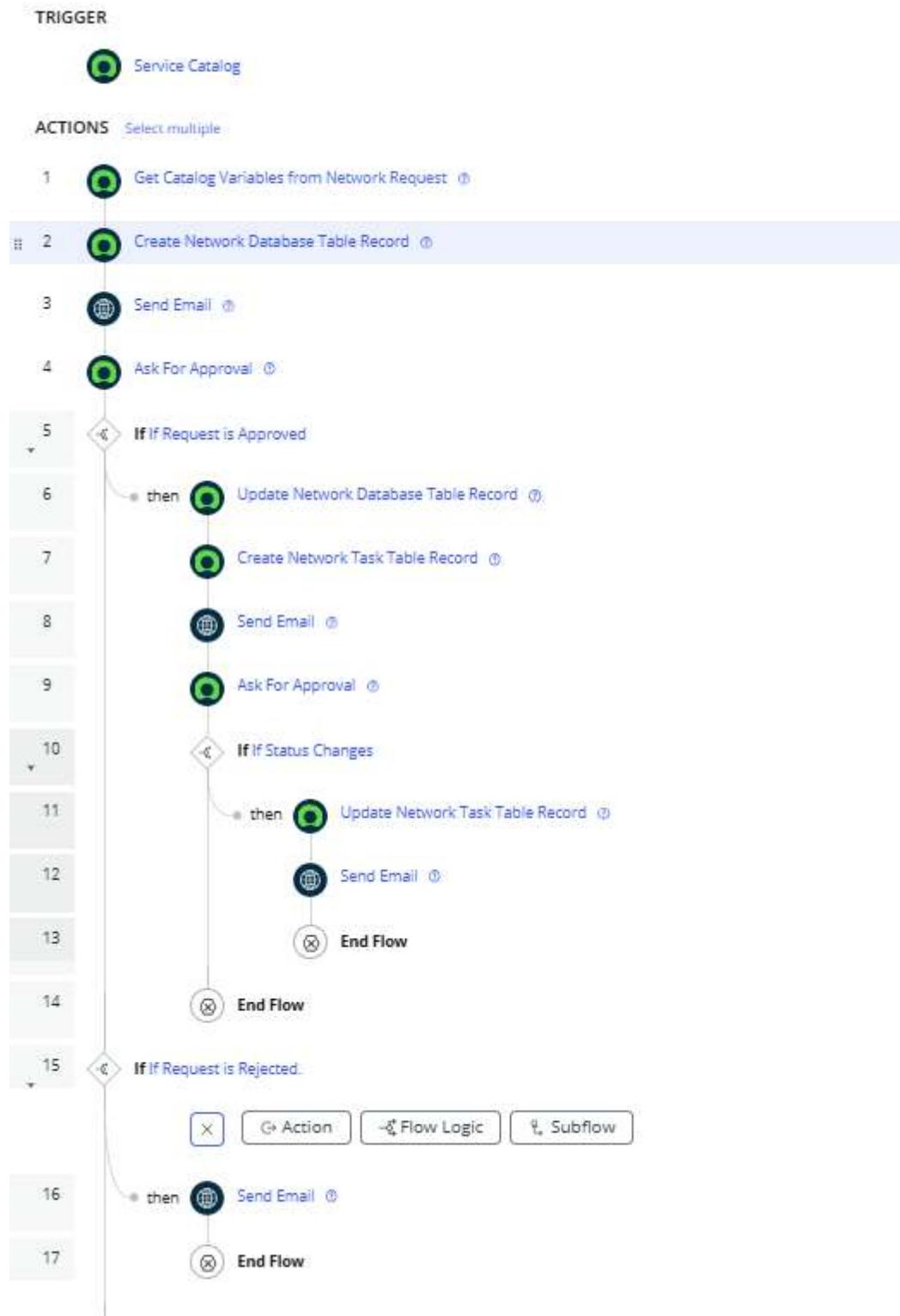
1. Add another If Condition for *Approval State is Rejected*.
2. In the Then branch:

Send Email (Rejected)

- Notify requestor that their request was rejected.
- Optionally include rejection comments.



OVERALL FLOW:



Summary

This project delivers an efficient ServiceNow-based solution for handling network service requests. By using a dedicated service catalog, automated approval workflows, and real-time notifications, it streamlines the request process for both users and technicians. The system ensures accurate request capture, faster resolution through automation, and better visibility with reporting and SLA tracking.

By KOTA ASHA