



NetBrain® Integrated Edition 7.1
Quick Start Guide (CM)

1. Managing Network Changes	3
1.1. Creating a Network Change Task	4
1.1.1. Defining the Change Node.....	5
1.1.2. Defining the Benchmark Before Node.....	6
1.1.3. Defining the Execute Node	6
1.1.4. Defining the Approval Node.....	7
1.1.5. Defining the Benchmark After Node	15
1.1.6. Defining the Compare Node	15
1.1.7. Adding and Defining a Note Node	15
1.2. Approving a Network Change Task.....	16
1.3. Executing a Network Change Task.....	18
1.3.1. Executing a Network Change Task on Demand	18
1.3.2. Executing a Network Change Task on Schedule	21
1.4. Archiving a Network Change Task.....	22
1.5. Browsing Change Tasks.....	23
1.6. Network Change Templates	25
1.7. Managing Configuration Templates	27
1.8. Managing Network Change Credentials	28
1.9. Access Controls	28

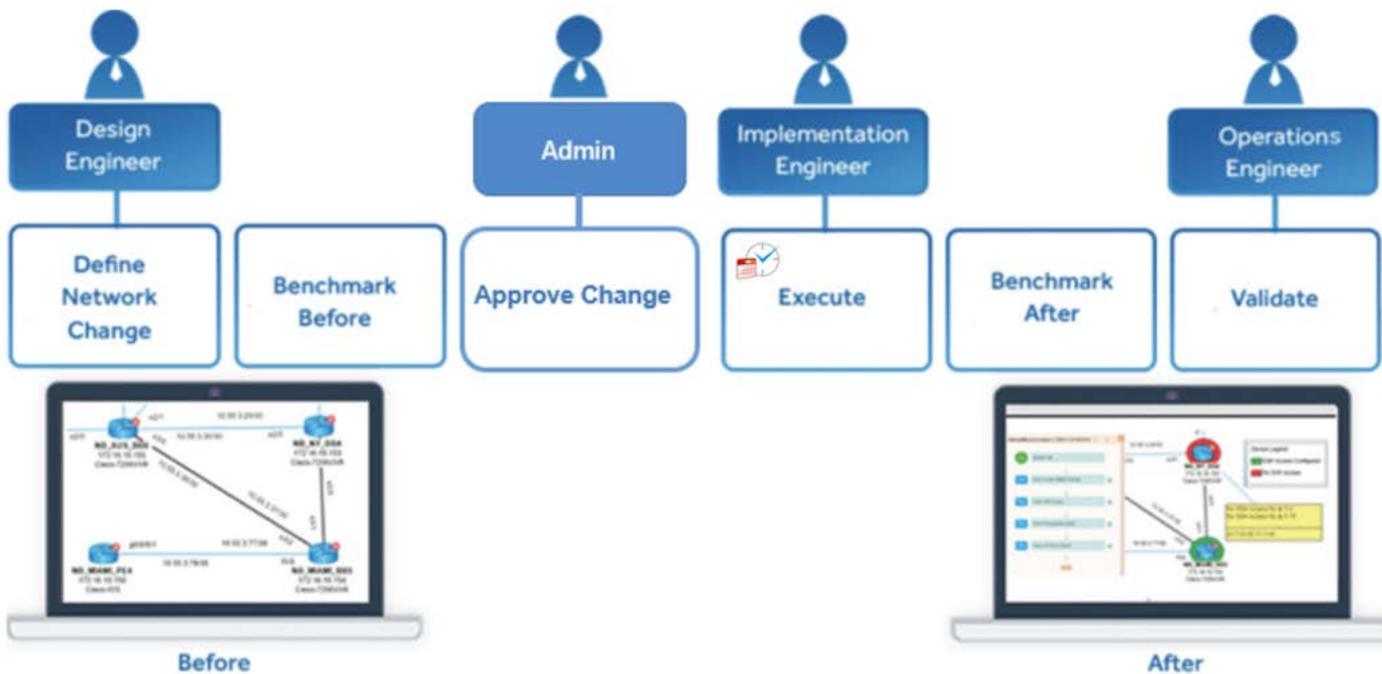
1. Managing Network Changes

Managing network changes can be a challenging task for network engineers during regular maintenance, which is prone to human errors and requires seamlessly planning, implementation and verification. Based on the executable runbook automation framework, NetBrain Network Change Management feature introduces an adaptive workflow to document the best practices of change management process and ensures a safer network change task:

- **Reduce Human Error in Network Changes** — the Design, Implementation, and Operations teams can leverage a single map to define and validate changes. Changes can even be deployed with automation.
- **Implement Changes with One Click** — the Design team can pre-define a configuration template which the Implementation team can execute with one click. If required, these changes can be rolled back easily.
- **Validate Changes Visually** — the Operations team can verify whether the changes were implemented as-planned by executing a pre-defined runbook, without adverse impact.

Network Change Task Flow

The typical workflow of a Change Management task is as follows:



1. [Define Network Change](#) — define network change, such as commands and devices.
2. [Benchmark Before](#) — collect network data as a baseline before implementing the change.
3. [Approve Network Change](#) — approve a network change task in NetBrain system or an external system.

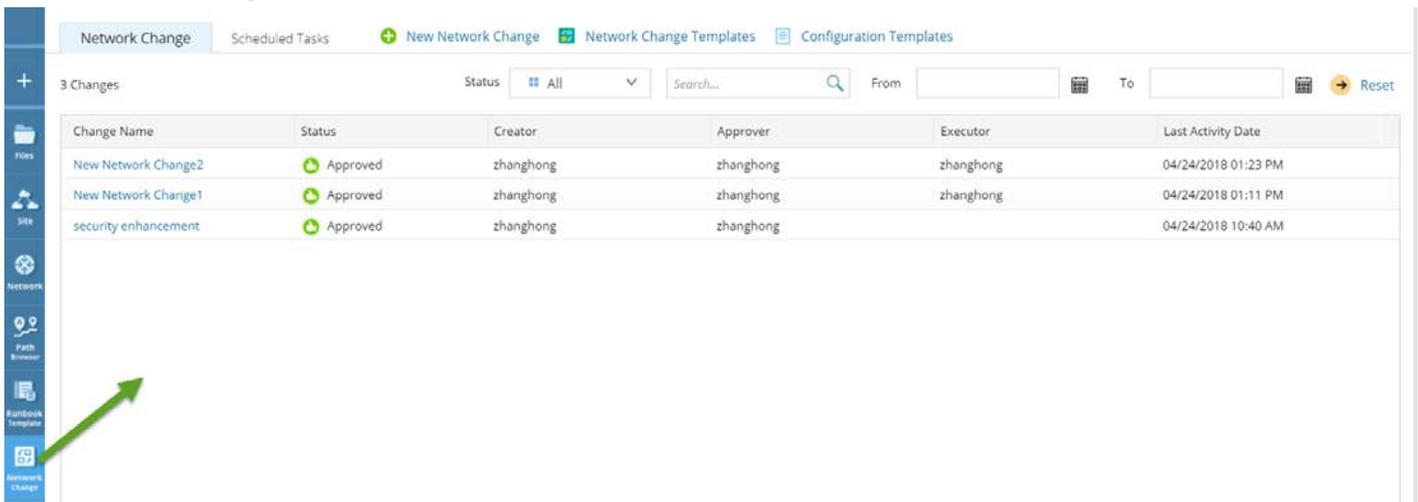
4. [Execute Network Change](#) — execute commands on devices.
5. [Benchmark After](#) — collect network data after implementing the change.
6. [Compare](#) — compare and validate the differences between two benchmark baselines.

1.1. Creating a Network Change Task

This section introduces how to create a network change task as well as the nodes and their functions involved in the network change task.

Tip: Before creating a network change task, ensure that you have sufficient privilege. See [Access Controls](#) for more details.

1. Click **Network Change** on the taskbar.



2. In the **Network Change** pane, click **New Change**.

Tip: The **Network Change** pane displays all network change tasks created in a domain.

3. In the **New Network Change** dialog, enter the task name, select a template and click **OK**. The Runbook pane opens. Then you can define the network change task node by node in the runbook.
 - [Defining the Change node](#)
 - [Defining the Benchmark Before node](#)
 - [Defining the Execute Node](#)
 - [Defining the Approval Node](#)
 - [Defining the Benchmark After node](#)

- [Defining the Compare node](#)
- [Adding and Defining a Change Note Node](#)

4. Click the  icon to save the map. The network change task is saved in the map.

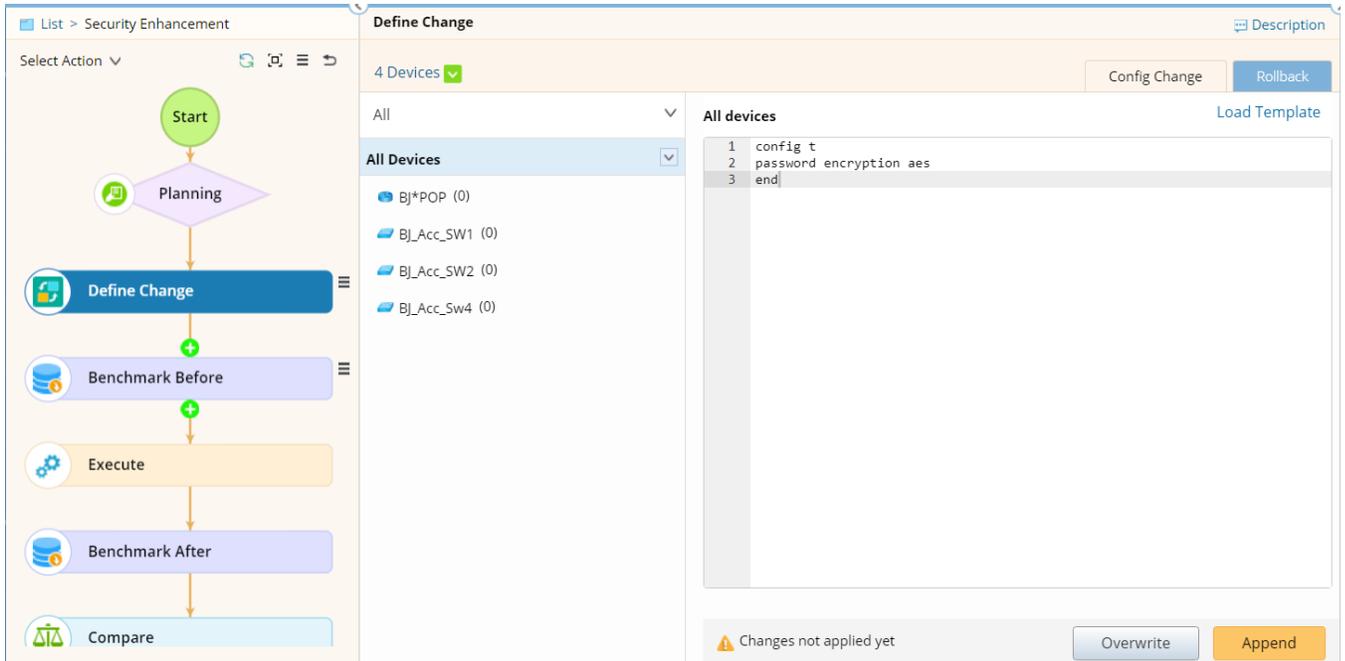
1.1.1. Defining the Change Node

The **Define Change** node is to define the target devices and commands that will be pushed to the target devices.

1. Select the **Define Change** node in the runbook.
2. In the **Define Change** pane, define the following settings:
 - 1) Click the  icon to select the target devices. By default, all devices on the map will be selected.

Tip: Point to a device and click the  icon to select more options.

- 2) In the **Config Change** tab, enter the commands that you want to execute on the target devices.



Example: Enter the commands to encrypt a password.

```
config t
password encryption aes
end
```

Tip: You can click **Load Template** and select a predefined configuration template. To create a new configuration template, see [Managing Configuration Templates](#) for details.

3) Select **All Devices** and click **Append** to add the command to all devices.

Tip: To clear the applied change command to all devices, select **All Devices** and click **Overwrite**.

4) (Optional) To cancel a specific change, click the **Rollback** tab and enter the rollback commands.

1.1.2. Defining the Benchmark Before Node

The **Benchmark Before** node is used to store the live data of the involved devices before the task implementation. Additionally, the data will be used as the source for data comparison before and after executing the change management task.

1. Select the **Benchmark Before** node in the runbook.
2. In the **Benchmark Before** pane, define the following settings:
 - 1) Click the  icon to select the target devices for the benchmark.
 - 2) Add CLI commands by using either way:
 - Manually enter a CLI command and click **Add**. You can add multiple commands.
 - Click the  icon to select **Load CLI Templates** to batch add commands.
 - 3) Click **Select Data** to select data types that you want to retrieve.

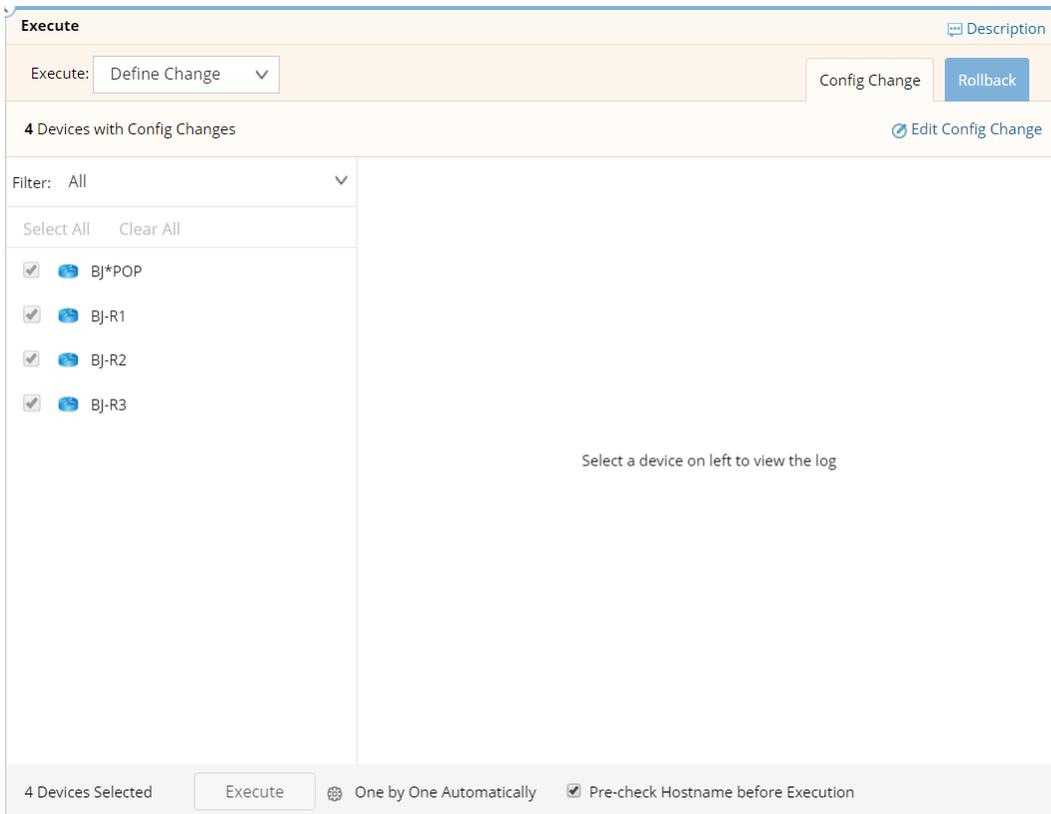
1.1.3. Defining the Execute Node

The **Execute** node is used to implement one of the defined changes on devices.

Tip: The **Execute** node cannot be executed unless the change management task is approved. See [Defining the Approval Node](#) to submit the approval request.

1. Select the **Execute** node in the runbook.

2. In the **Execute** pane, define the following settings:



- 1) Select a change node from the drop-down list. The related devices and change commands will be loaded.
- 2) Click the  icon next to the **Execute** button to set the execution methods:
 - **One by One Automatically** — push change commands to a device, and then push them to the next device automatically after a brief pause.
 - **In Batches Automatically** — push change commands to a batch of devices, and then push them to the next batch of devices automatically after a brief pause.
 - **One by One Manually** — push change commands to a device manually, and you need to click **Execute** to push commands to the next device manually.
 - **Sleep Time between all command** — the time interval of sending CLI commands. By default, the interval is 2 seconds.

1.1.4. Defining the Approval Node

The **Approval** node is used to define the approval request to execute the change.

You can define two types of approval requests:

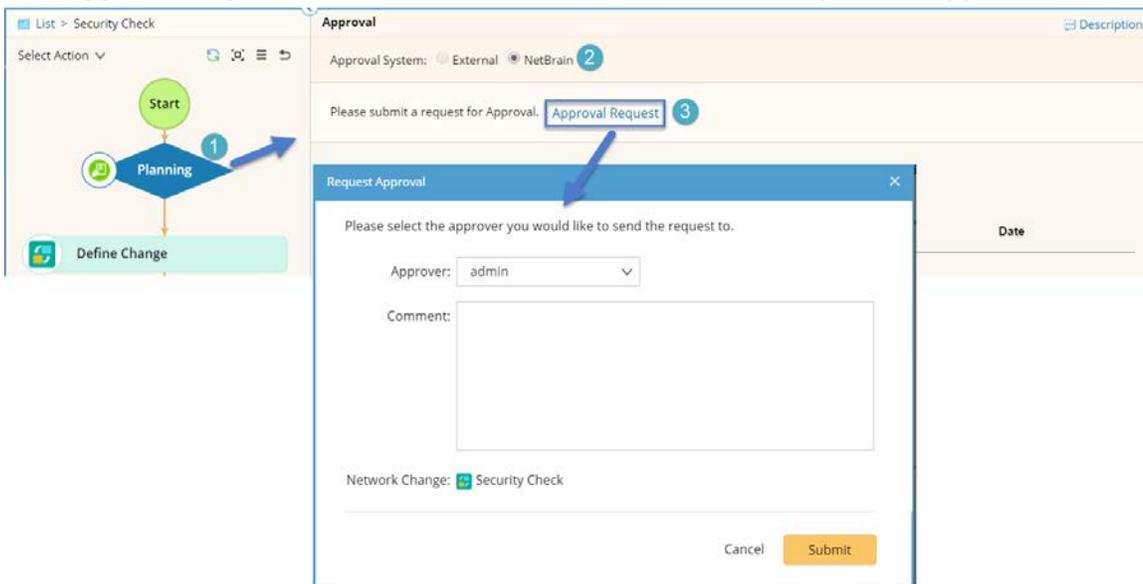
- [NetBrain](#) — define and complete an approval workflow in NetBrain System.

- [External](#) — integrate with an external ticket system and approve the task in the external system.

Tip: Before using this feature, ensure that you have sufficient privilege. See [Access Controls](#) for more details.

Defining a NetBrain Approval Request

1. Select the **Planning** node in the runbook.
2. In the **Approval** pane, select **NetBrain** in the **Approval System** field.
3. Click **Approval Request** and adjust the following settings in the **Request for Approval** dialog.

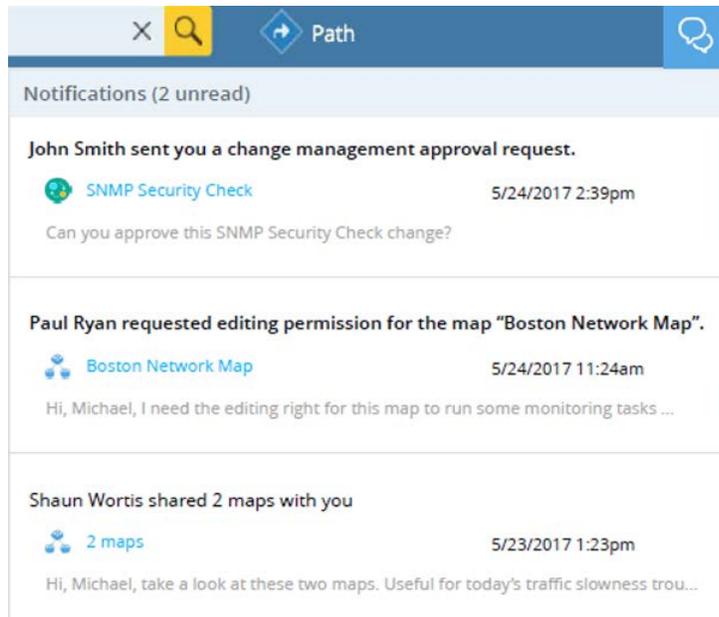


- 1) Select an approver from the **Approver** drop-down list.

Tip: If the user you want to select is not on the list, contact your NetBrain administrator to assign the required privilege to the user.

- 2) In the **Comment** field, enter some comments about the approval request.

- 3) Click **Submit**. After the request is submitted, the designated approver will receive an email (if the email setting is configured) and system notification.



Defining an External Approval Request

1. Select the **Planning** node in the runbook.
2. In the **Approval** pane, select **External** in the **Approval System** field.

Note: If the **External** option is disabled, see [Approval Settings](#) to adjust the approval option.

3. Click **Copy Network Change ID**. The ID will be used to link with a ticket in an external system, thus building a connection with the **Approval** node in the external system for status synchronization.
4. Create a change ticket in the external system and customize two fields in the ticket to save the Network Change ID and returned NetBrain URL.
5. Create a business rule to define ticket and network change task status mapping and API calls in the external system. The rule mainly covers the following components:
 - How to access your NetBrain domain.
 - Map ticket state and Change Management status.
 - Bind ticket ID to Change Management ID.
 - Call NetBrain API to Set Change Management Status when ticket state changes.

Example: Define Status Mapping and API Calls in ServiceNow.

```
/**
 * ServiceNow Script Example, Please follow this script
```

```

*
*/
(function executeRule(current, previous /*null when async*/) {

    var NB_API_PREFIX = 'http://your web API server address/ServicesAPI/API/V1';

    // login to get token & logout
    var tokenEndPoint = NB_API_PREFIX + 'Session';
    var tenantDomainEndPoint = NB_API_PREFIX + 'Session/CurrentDomain';

    // credential to log into netbrain
    var credential = {
        username: 'admin',
        password: 'admin'
    };
    var tenantDomain = { tenantId: 'xxx', domainId: 'xxx' };

    // api time out default is 10 seconds
    var API_TIME_OUT = 10 * 1000;

    // store the netbrain api token
    var nbToken = '';

    // *****
    var vendor = 'ServiceNow';
    var stateMapping = {
        'not requested': 0, // planning
        'requested': 1, // pending
        'approved': 2,
        'rejected': 3
    };

    /**
     * Request the token
     * @return hasToken: bool
     *
     */
    function login() {
        var request = newRequest(tokenEndPoint, 'POST', credential);
        try {
            gs.info('send request to netbrain, get the token'); //
            var response = request.execute();
            var responseBody = response.getBody();
            var obj = new JSON.parse(responseBody);
            nbToken = obj.token;
            return nbToken.length > 0;
        } catch (ex) {
            // todo handle get token failed exception where to check this error msg
            gs.error('get netbrain token failed. ');
            gs.error(ex.getMessage());
            return false;
        }
        return false;
    }
}

```

```

/**
 * logout netbrain, remove token
 * @return
 * @constructor
 */
function logout() {
    var body = {
        token: nbToken
    };
    var request = newRequest(tokenEndPoint, 'DELETE', body);
    try {
        gs.info('logout netbrain.');
```

```

        request.execute();
        gs.info('logout netbrain success.');
```

```

    } catch (ex) {
        gs.error('get netbrain token failed.');
```

```

        gs.error(ex.getMessage()); // should retry?
    }
}

function addActivity(message) {
    current.work_notes = message;
    current.update();
}

/**
 * Request to set tenant & domain
 * @return success: bool
 *
 */
function setTenantDomain() {
    var request = newRequest(tenantDomainEndPoint, 'PUT', tenantDomain);
    try {
        gs.info('set tenant & domain info.');
```

```

        var response = request.execute();
        var responseBody = response.getBody();
        var obj = new JSON.parse(responseBody);
        return true;
    } catch (ex) {
        gs.error('set tenant & domain failed.');
```

```

        gs.error(ex.getMessage()); // should retry?
        return false;
    }
    return false;
}

/**
 * make a http call to netbrain to bind change request id to a cm runbook id
 * @return
 *
 */
function apiCMBinding() {
    var ticketId = current.getValue('sys_id');
```

```

    var runbookId = current.getValue('netbrain_cm_id');// this might be different for

```

```

different customer
    var ticketName = current.getValue('number'); // this might be different for different
customer
    var endPoint = NB_API_PREFIX + 'CM/Binding';
    var snState = current.getValue('approval');
    if (stateMapping[snState] == null) {
        gs.error('cannot find the correspond state from netbrain.');
```

return;

```

    }
    var postData = {
        runbookId: runbookId,
        ticketId: ticketId,
        vendor: vendor,
        ticketName: ticketName,
        state: stateMapping[snState]
    };
    var request = newRequest(endPoint, 'POST', postData);
    try {
        gs.info('send change management binding request.');
```

var response = request.execute();

```

        var responseBody = response.getBody();
        var obj = new JSON.parse(responseBody);
        if (obj.statusCode == 0) {
            gs.info('change management binding to ' + runbookId + ' success.');
```

addActivity('change management binding to ' + runbookId + ' success.');

```

        } else {
            gs.info(obj.statusDescription);
            // please change it to a proper msg, and this message is going to display on
service now page
            gs.addErrorMessage('binding Failed: ' + obj.statusDescription);
            addActivity('binding Failed: ' + obj.statusDescription);
        }

    } catch (ex) {
        gs.error('change management binding failed.');
```

gs.error(ex.getMessage()); // should retry?

```

        gs.addErrorMessage('binding Failed: ' + ex.getMessage());
        addActivity('binding Failed: ' + ex.getMessage());
    }
}

/**
 * make a http call to netbrain to set cm runbook approval state
 */
function apiUpdateCMApprovalState() {
    var ticketId = current.getValue('sys_id'); // this might be different for different
customer
    var runbookId = current.getValue('netbrain_cm_id');// this might be different for
different customer
    var endPoint = NB_API_PREFIX + 'CM/approval/state';
    var ticketName = current.getValue('number'); // this might be different for different
customer
    var snState = current.getValue('approval');
    if (stateMapping[snState] == null) {
        gs.error('cannot find the correspond state from netbrain.');
```

```

        return;
    }
    var postData = {
        runbookId: runbookId,
        ticketId: ticketId,
        vendor: vendor,
        ticketName: ticketName,
        state: stateMapping[snState]
    };
    var request = newRequest(endPoint, 'POST', postData);
    try {
        gs.info('set change management state.');
```

success.'

```

        var response = request.execute();
        var responseBody = response.getBody();
        var obj = new JSON.parse(responseBody);
        if (obj.statusCode == 0) {
            var msg = 'set change management(' + runbookId + ') to state ' + snState + '
            gs.info(msg);
            addActivity(msg);
        } else {
            gs.info(obj.statusDescription);
            gs.addErrorMessage('update approval failed: ' + obj.statusDescription);
            addActivity('update approval failed: ' + obj.statusDescription);
        }
    } catch (ex) {
        gs.error('set change management state failed.');
```

```

        gs.error(ex.getMessage());
        gs.addErrorMessage('Error: ' + ex.getMessage());
        addActivity('Error: ' + ex.getMessage());
    }
}

function newRequest(url, method, data) {
    var request = new sn_ws.RESTMessageV2();
    request.setEndpoint(url);
    request.setHttpMethod(method);
    request.setRequestBody(JSON.stringify(data));
    request.setRequestHeader('Accept', 'application/json');
    request.setRequestHeader('Content-Type', 'application/json');
    request.setRequestHeader('token', nbToken);
    request.setHttpTimeout(API_TIME_OUT);
    return request;
}

/**
 * update a value of service now
 */
function updateColumnValue(table, primaryKey, primaryVal, targetColumn, targetVal) {
    var gr = new GlideRecord(table);
    gr.addQuery(primaryKey, primaryVal);
    gr.query();
    if (gr.next()) {
        gr.setValue(targetColumn, targetVal);
    }
}

```

```

        gr.update();
    }
}

/**
 * make a http call to netbrain to binding change request id to a cm runbook id
 */
function main() {
    var hasToken = login();
    if (hasToken) {
        gs.debug('login success. ');
        if (setTenantDomain()) {
            // business logic
            apiCMBinding();
            apiUpdateCMAApprovalState();
        }
        logout();
    } else {
        gs.info('login failed, quit now...');
    }
}

main(); // run
})(current, previous);

```

Approval Settings

To set the approval type for network change management:

1. Log in to Domain Management page.
2. In the Domain Management page, click **Operations > Domain Maintenance > Change Management Settings** from the quick access toolbar.
3. On the **Change Management Settings** tab, enable the approval method you want to use.

The screenshot shows a web interface with two tabs: 'Start Page' and 'Change Management Settings'. Under the 'Change Management Settings' tab, there is a checkbox labeled 'Enable Network Change Function for Current Domain' which is checked. Below this, under the heading 'Approval Settings', there are two more checkboxes: 'External Approval System' and 'Netbrain Approval System', both of which are also checked. At the bottom of the settings area is an orange 'Submit' button.

4. Click **Submit**.

1.1.5. Defining the Benchmark After Node

The **Benchmark After** node is used to collect network data after the change is implemented, thus verify if the change has been deployed successfully. Additionally, the benchmark results will be used to compare with the data collected in the **Benchmark Before** node.

1. Select the **Benchmark After** node in the runbook.
2. In the **Benchmark After** pane, define the following settings:
 - 1) Click the  icon to select the target devices for the benchmark.
 - 2) Add CLI commands by using either way:
 - Manually enter a CLI command and click **Add**. You can add multiple commands.
 - Click the  icon to select **Load CLI Templates** to batch add commands.
 - 3) Click **Select Data** to select data types that you want to retrieve.

1.1.6. Defining the Compare Node

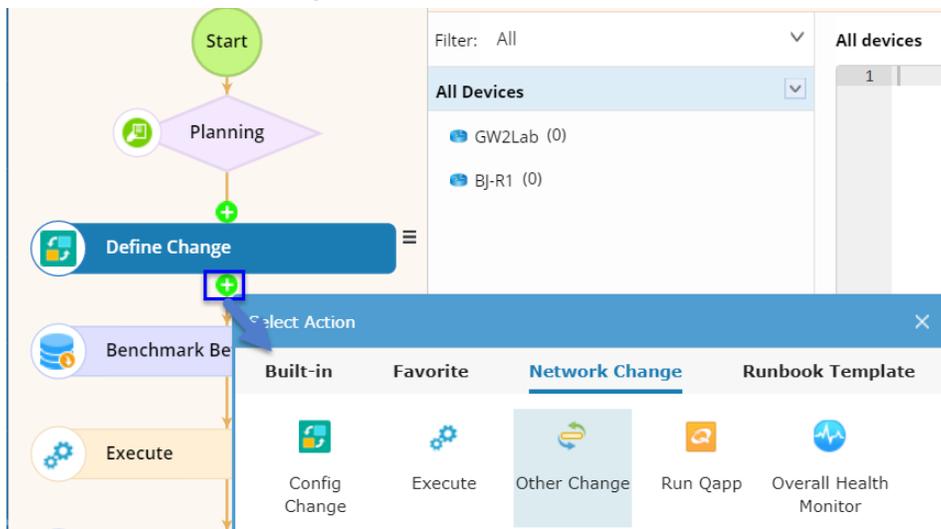
The **Compare** node is used to compare the data between [Benchmark before](#) and [Benchmark after](#) to verify the changes.

When you define or edit a **Compare** node, no settings are required. Select data source for comparison when you [execute the Compare node](#).

1.1.7. Adding and Defining a Note Node

The **Other Change** node provides note function to remind the change executor to do some operations, such as discovering new devices to the domain, removing devices from the domain and updating modules beyond the change management task.

1. Point to the **Define Change** node and click the  icon.



2. In the **Select Action** dialog, select **Other Change** on the **Network Change** tab.
3. Enter the descriptions in the **Description** field.

Tip: You can mark the **Other Change** node as **Completed** when it is approved. Then it will be locked and cannot be edited anymore.

1.2. Approving a Network Change Task

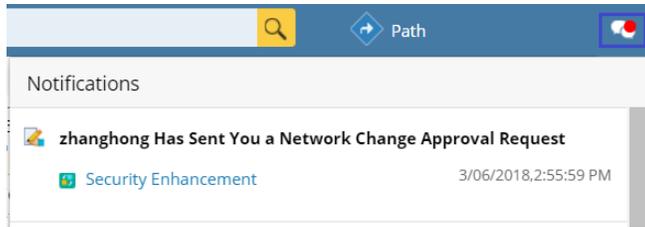
When a network change task is created, it needs to be approved before the implementation. Select to approve a network change task based on the way that the task creator submits the approval request:

- [Approving a Network Change in NetBrain System](#)
- [Approving a Network Change in an External System](#)

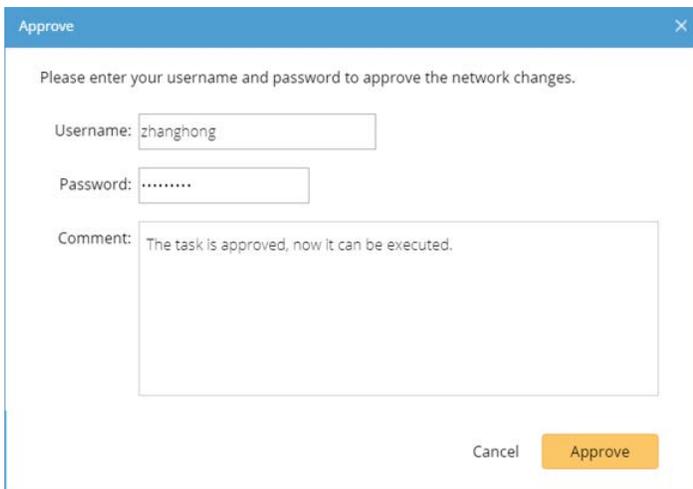
Approving a Network Change in NetBrain System

If a task creator selects the way to approve in NetBrain system, the designated approver will receive an email and system notification after the approval request is submitted.

Example: Approve an **SNMP Security Check Change** Request.



1. Click the **SNMP Security Check** hyperlink in the notification list.
2. In the opened runbook, click the **Pending** node.
3. In the **Approval** pane, click **Approve**.
4. In the pop-up dialog, enter the username and password of the approver account, and click **Approve**.



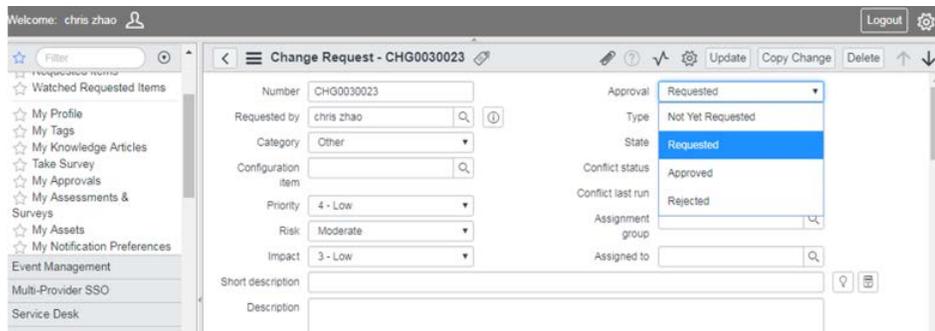
5. Click **Yes** to confirm the approval in the **Confirmation** dialog.

The creator who submits the request will receive an email and system notification about the approval status change.

Approving a Network Change in an External System

After a ticket is created and associated with the network change task in an external system, the change task status will be triggered to synchronize when you change the ticket status.

Example: A ticket state associated with the network change status in ServiceNow.



For how to integrate a change management task with an external ticket system for approval, see [Defining an External Request](#) for details.

1.3. Executing a Network Change Task

After a network change task is approved, users with the required privileges can execute the network change task on demand or on a schedule.

- [Executing a Network Change Task on Demand](#)
- [Executing a Network Change Task on Schedule](#)

1.3.1. Executing a Network Change Task on Demand

This section introduces how to manually execute a network change task based on the following typical flow:

1. [Executing Benchmark Before Change](#) — collect the live data of the involved devices before the network change.
2. [Implementing Change](#) — implement the network changes.
3. [Executing Benchmark After Change](#) — collect the live data of the involved devices after the network change.
4. [Comparing Before and After](#) — compare the differences before and after the network change.

1.3.1.1. Executing Benchmark Before Network Change

To minimize the risk of implementing a network change, it is recommended to execute a benchmark task to back up the live data of the involved devices before the change. Additionally, the data will be used as the source for comparison and validation.

1. Click the **Benchmark Before** node in the runbook.
2. In the **Benchmark Before** pane, click the  icon to customize devices and define data type respectively if necessary. The predefined devices, CLI commands, and data types will be used by default.
3. Click **Run** to benchmark data. Click **Yes** to save the map when the **Confirmation** dialog prompts.

Tip: If you select the **Auto update all selected data in Current Baseline**, the configuration files stored in the Current Baseline will be overwritten with the latest retrieved data.

4. Click the **Results** tab and select a device to view its results.
5. Click the **Execution Log** tab and select a device to view its logs.

1.3.1.2. Implementing Network Change

Before implementing the change, it is recommended to make the following preparations:

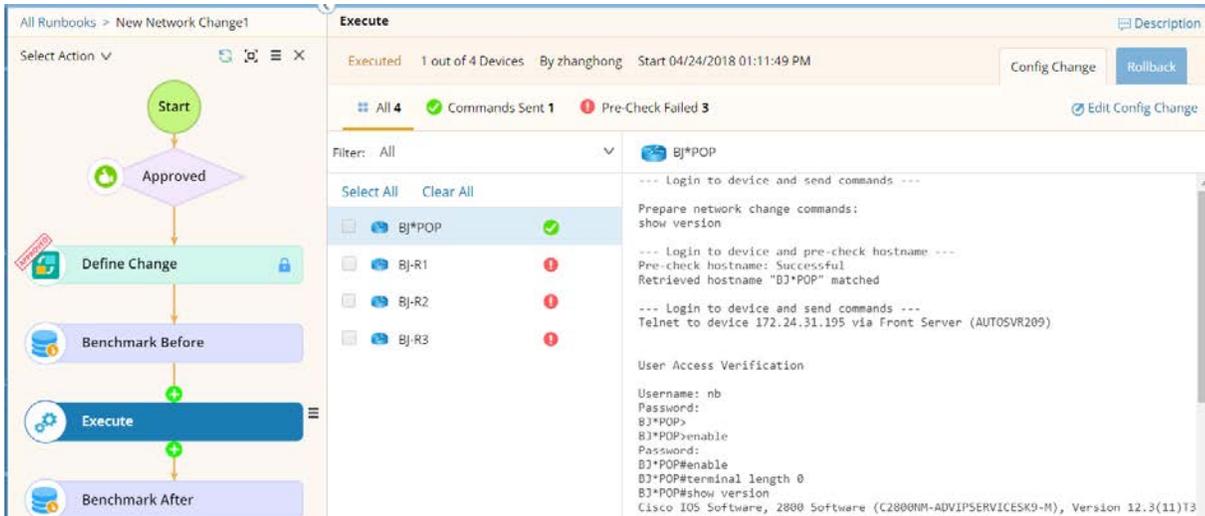
- Set access credentials to log into devices. See [Managing Network Change Credentials](#) for details.
- Tune devices to check their reachability: click the **Define Change** node > drop-down list of **All Devices** > select **Tune**.

Proceed to the following steps to implement a network change task.

1. Click the **Execute** node in the runbook.
2. In the **Execute** pane, click the  icon to define the execution options.
 - **One by One Automatically** — push change commands to a device, and then push them to the next device automatically after a brief pause.
 - **In Batches Automatically** — push change commands to a batch of devices, and then push them to the next batch of devices automatically after a brief pause.
 - **One by One Manually** — push change commands to a device manually, and you need to click **Execute** to push commands to the next device manually.
 - **Sleep Time between all commands** — the time interval of sending CLI commands. By default, the interval is 2 seconds.
3. Check the **Pre-check Hostname before Execution** option.

This option checks whether the target devices that you will push commands to are still the same as planned before sending change commands to devices. If a device hostname changed or retrieving hostname failed, skip the device to push commands.

4. Click **Execute**. Read the notices and click **OK** when the **WARNING** dialog prompts.



- To roll back the config change, select the **Rollback** tab and then click **Execute** to execute the rollback config.
- If you select **One by One Manually** in step 2, click **Continue** to execute the change on the next device.
- When the execution of a device is done, an execution status icon will be displayed on the device. The following table describes the execution status in detail:

Execution status	Description
Commands Sent	All commands have been sent to devices.
Incomplete	Parts of commands have been sent to devices.
Pre-Check Failed	Failed to retrieve device hostname.
Login Failed	The system fails to remotely connect to the device using Telnet and SSH.

1.3.1.3. Executing Benchmark After Network Change

To validate the network change, it is recommended to collect the network data after the network change and use the data for further comparison.

1. Click the **Benchmark After** node.
2. In the **Benchmark After** pane, click the  icon to customize devices and define data type respectively if necessary. The predefined devices, CLI commands, and data types are used by default.
3. Click **Run** to benchmark data.

Tip: If you select the **Auto update all selected data in Current Baseline** check box, the configuration files stored in the **Current Baseline** will be overwritten with the retrieved data.

4. Click the **Results** tab and select a device to view its results.
5. Click the **Execution Log** tab and select a device to view its logs.

1.3.1.4. Comparing After and Before

To verify whether the network changes go as expected, you can use the **Compare** node to validate the network differences after and before the network change.

1. Click the **Compare** node in the runbook.
2. In the **Compare** pane, click the  icon to select the target devices. By default, all devices on the map page are selected.
3. Select the **Benchmark After** and **Benchmark Before** respectively in the **Dataset1** and **Dataset2** fields.
4. Click **Compare**.

1.3.2. Executing a Network Change Task on Schedule

You can set a scheduled time point for a change management task. Then all actions will be executed automatically and sequentially when it reaches the time point.

1. Open the target change management task, and click the  icon to select **Schedule**.
2. In the **Schedule** dialog, select the time point to execute the change:
 - **Execution Time** — the time to start the task.
 - **Do not Execute After** — the time to drop the task if it does not start before the time.

3. Click **Save**.

The screenshot shows the 'Schedule' dialog box in the NetBrain interface. The dialog is titled 'Schedule' and contains the following information:

- Network Change: New Network Change1
- Execution time: 05/21/2018 10:37 AM
- Do Not Execute After: 05/22/2018 3:37 AM

Below the input fields, there is a note: "There might be a system delay due to the backlog of tasks. In the event of a system delay, please enter the time you want to abort the execution." At the bottom of the dialog, there are two buttons: "Cancel" and "Save".

1.4. Archiving a Network Change Task

When a change task is completed, you can archive it. Once archived, a change task cannot be edited anymore. But you can still open and view history results of it.

1. Open the target change management task, click the  icon and select **Archive Change** from the drop-down menu.
2. In the **Archive** dialog, click **Archive**.

The screenshot shows the 'Archive' dialog box in the NetBrain interface. The dialog is titled 'Archive' and contains the following information:

- Archiving will lock all functions in this Runbook.
- Are you sure you want to proceed?
- You can make changes by creating a copy of this Runbook.

At the bottom of the dialog, there are two buttons: "Cancel" and "Archive".

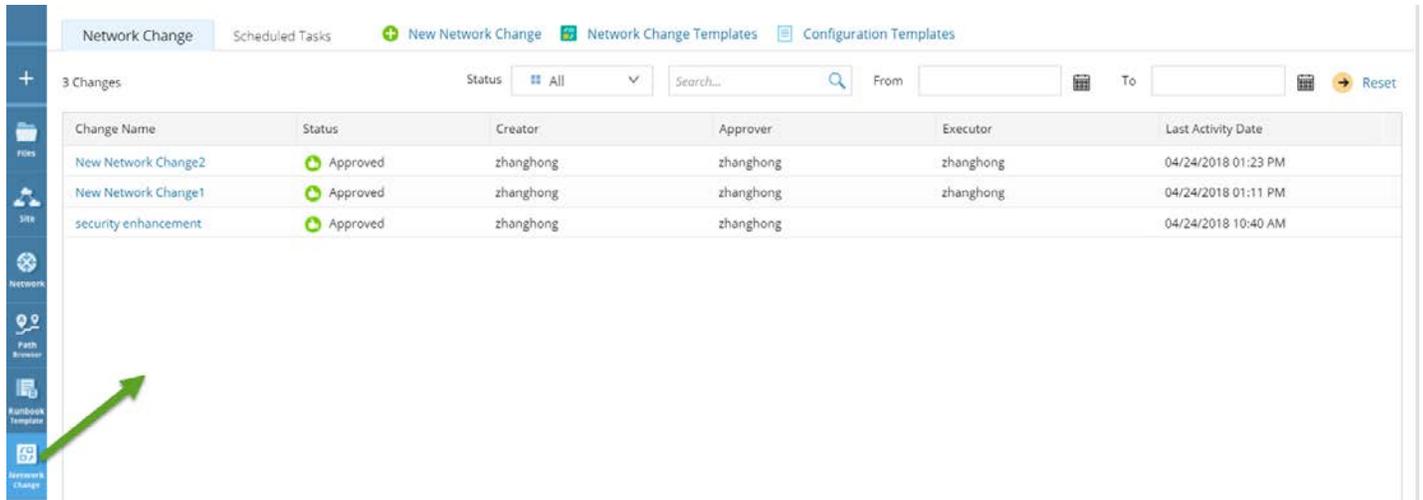
1.5. Browsing Change Tasks

In the Network Change pane, you can view and manage all network change tasks in your domain.

Viewing Network Change Tasks in Your Domain

Follow the steps below to view network change tasks in your domain:

1. Click **Network Change** on the taskbar.



2. In the **Network Change** pane, view the network change tasks. Each task includes the following information:

Field	Description
Change Name	The name of a network change task.
Status	The status of a network change task. It includes five types of status: <ul style="list-style-type: none">▪ Planning — indicates that the task is in planning and not sent for approval yet.▪ Pending — indicates that a network change is waiting for approval.▪ Approved — indicates that a network change is approved by an approver.▪ Rejected — indicates that a network change is rejected.▪ Archived — indicates that a network change is archived.
Creator	The creator of a network change task.
Approver	The approver of a network change task
Executor	The executor of a network change task
Last Activity Date	The date when a network change task is last executed.

3. Click the **Scheduled Tasks** tab to view the information of the scheduled tasks in your domain. A scheduled task has 6 types of status.

- **Waiting** — indicates that the task awaits to be executed. It cannot be edited.
- **Running** — indicates that the task is being executed.
- **Stopped** — indicates that the task is stopped manually.
- **Executed** — indicates that the task is executed.
- **Aborted** — indicates that a scheduled task is aborted because it does not start before the specified time.
- **Stopping** — indicates that the task is being stopped.

Finding a Network Change Task

You can quickly find a network change task via the following methods:

- **Search by Keywords** — enter a keyword in the search bar and press the **Enter** key on your keyboard to start searching.
- **Filter by Time** — select the start date and end date respectively, and click the  icon to start filtering.
- **Filter by Status** — click the **Status** drop-down list and select a status. Network change tasks will be automatically filtered by the selected status.

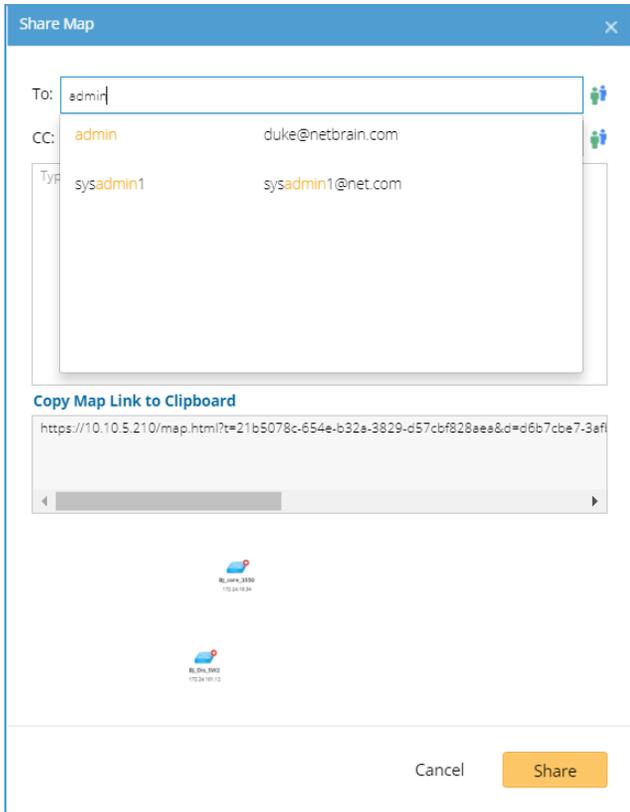
Tip: To reset the criteria for search or filter, click **Reset**.

Sharing a Network Change Task

You can also share a network change task as follows:

1. Right-click a task entry and select **Share** from the right-click menu.
2. In the **Share Map** dialog, enter the email addresses in the **To** field, or click the **To** field and select a user from the drop-down menu.

3. Click **Share**. The task will be shared via the map link.



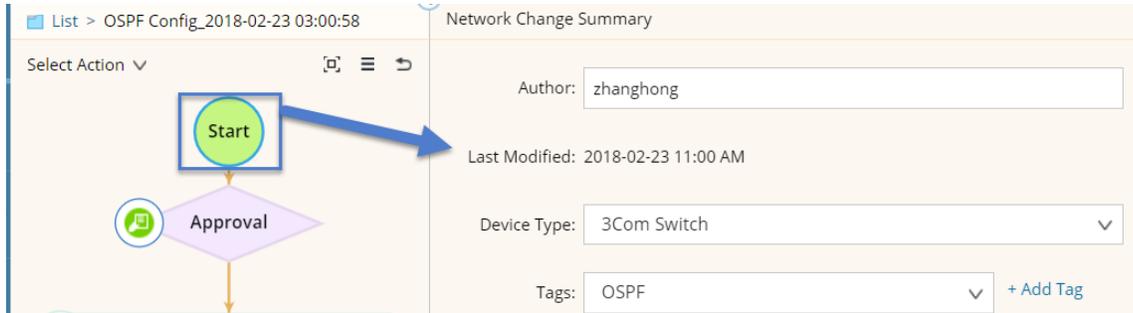
1.6. Network Change Templates

You can save frequently used network changes tasks as templates so that you can quickly create other network change tasks based on the predefined templates.

To browse network change templates in your domain:

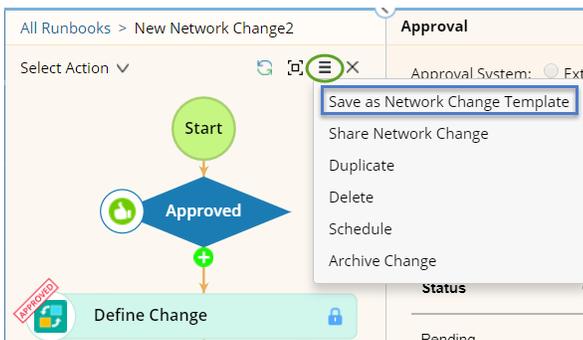
1. Click **Network Change** in the taskbar.
2. In the Network Change Manager, click **Network Change Templates**. The system provides the following three categories of change templates:
 - **Built-in Templates** — contains the pre-installed change templates. This category is accessible to all users but can only be modified by system and tenant administrators.
 - **Shared Templates** — contains the change templates that are accessible to all users in this domain.
 - **My Templates** — contains the change templates that are only accessible to yourself in this domain.
3. To quickly find your interested change template, use the one or a combination of the following filters:

- **Device Type** — filter network change templates by specified device type (only if it has been defined in the **Start** node of a network change template).
- **Author** — filter network change templates by specified author name.
- **Tag** — filter network change templates by specified tag (only if it has been defined in the **Start** node of a network change template).

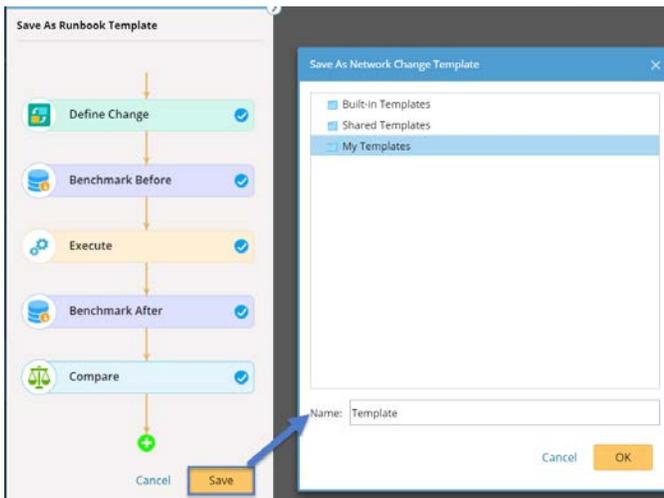


Creating a Network Change Template

1. [Create a network change task.](#)
2. In the Network Change Template editor, click the  icon and select **Save as Network Change Template**.



3. Click **Save**, select a location in the pop-up dialog, enter a name and click **OK**.

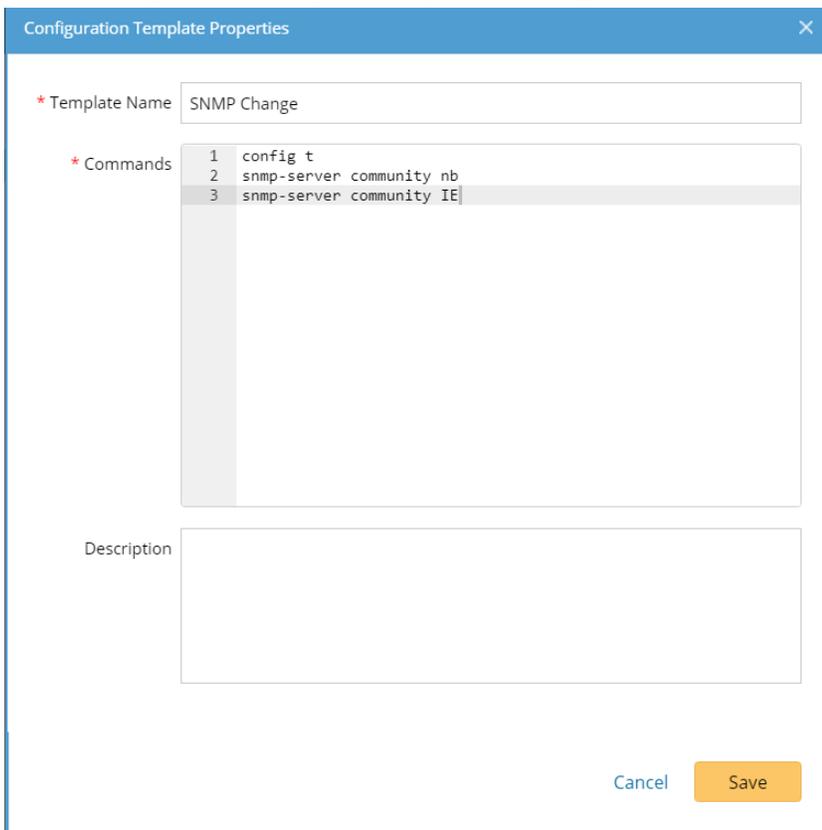


1.7. Managing Configuration Templates

To reuse configurations for network change tasks in the future, you can pre-define it as a config template. When [Defining the Change Node](#), you can directly load a config template instead of writing commands manually.

Proceed with the following steps to create a configuration template:

1. Click **Network Change** on the taskbar.
2. In the Network Change pane, click **Configuration Templates**.
3. In the **Manage Configuration Templates** pane, point to a folder and click the drop-down menu to select **New Template**.
4. In the **Configuration Template Properties** dialog, enter the template name and commands.



The screenshot shows a dialog box titled "Configuration Template Properties" with a close button (X) in the top right corner. The dialog contains the following fields:

- * Template Name:** A text input field containing "SNMP Change".
- * Commands:** A list box containing three entries:

1	config t
2	snmp-server community nb
3	snmp-server community IE
- Description:** An empty text input field.

At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

5. Click **Save** to complete the configuration template.

1.8. Managing Network Change Credentials

When users execute a change management task, the task will call and use the predefined credentials to access devices and issue relevant commands. Based on all possible credential policies, NetBrain offers multiple access modes and you can select a proper one according to your network credential mechanism.

To set the credential mode for network change management:

1. Log in to Domain Management page.
2. In the **Domain Management** page, click **Operations > Advanced Settings** from the quick access toolbar.
3. On the **Advanced Settings** tab, select an access credential mode:
 - **Use Private Credentials and shared Privilege Credentials** — use credentials defined in the Private CLI Settings to log in to devices and then use privilege credentials defined in the Shared Device Settings to enter the privilege mode. It is the default access credential mode.
 - **Use Private Credential and Private Privilege Credential** — use credentials in the Private CLI Settings to log in to devices and then use privilege credentials defined in the Private Device Settings to enter the privilege mode.
 - **Use Shared Credentials and Shared Privilege Credentials** — use credentials defined in the Shared Device Settings to log in to devices and then use privilege credentials defined in the Shared Device Settings to enter the privilege mode.

Note: Credentials in the Private CLI Settings are only available to the user account that creates these credentials. While credentials in the Shared Device Settings are available to all users who can access the domain.

1.9. Access Controls

The Change Management is an add-on feature which is only available on request. Contact [NetBrain Support Team](#) for further assistance.

Network Change Privileges

To ensure network security, only authorized users with the following privileges can create, approve or execute a network change task:

- **Create Network Change** — users with this privilege can create and edit a network change task. Domain admins and power users have this privilege by default.

- **Approve Network Change** — users with this privilege can approve a network change task. Domain admins and power users have this privilege by default.
- **Execute Network Change** — users with this privilege can execute a network change task. Domain admins and power users have this privilege by default.
- **View Network Change** — users with this privilege can open and view network change tasks in a domain, but cannot edit them. Domain admins and power users have this privilege by default.

Assigning Network Change Privileges to a Specific User

To assign privileges of change management to specific users, complete the following steps:

1. Log in to Domain Management page.
2. In the **Domain Management** page, click **Operations > Share Policy** from the quick access toolbar.
3. On the **Share Policy** tab, select a user in the left pane and check the **Network Change Creator/Executor/Approver** option based on your needs.
4. Click **Apply**.

