Walkthrough: Using Document Builder to create a custom runbook

PDF (systems-manager-ug.pdf#automation-walk-document-builder)

Kindle (https://www.amazon.com/dp/B078XCSXDS)

RSS (aws-systems-manager-user-guide-updates.rss)

The following walkthrough shows how to use Document Builder in the Systems Manager Automation console to create a custom runbook and then run the custom runbook.

The first step of the runbook you create runs a script to launch an Amazon Elastic Compute Cloud (EC2) instance. The second step runs another script to monitor for the instance status check to change to ok. Then, an overall status of Success is reported for the automation.

Before You Begin

Before you begin this walkthrough, do the following:

 Verify that you have administrator privileges, or that you have been granted the appropriate permissions to access Systems Manager in AWS Identity and Access Management (IAM).

For information, see Verifying user access for runbooks (./automation-setup.html#automation-setup-user-access).

• Verify that you have an IAM service role for Automation (also known as an *assume role*) in your AWS account. The role is required because this walkthrough uses the **aws:executeScript** action.

For information about creating this role, see Configuring a service role (assume role) access for automations (./automation-setup.html#automation-setup-configure-role).

For information about the IAM service role requirement for running **aws:executeScript**, see Permissions for using runbooks (./automation-document-script.html#execution-permissions).

Verify that you have permission to launch EC2 instances.

For information, see IAM and Amazon EC2 (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/UsingIAM.html#intro-to-iam) in the Amazon EC2 User Guide for Linux Instances.

Topics

- Step 1: Create the custom runbook (#automation-walk-document-builder-create)
- Step 2: Run the custom runbook (#automation-walk-document-builder-run)

Step 1: Create the custom runbook

Use the following procedure to create a custom runbook that launches an EC2 instance and waits for the instance status check to change to ok.

Tip

If you copy and paste values from this walkthrough into Document Builder, such as parameter names and handler names, make sure to delete any leading or trailing spaces added to the text value you enter.

To create a custom runbook using Document Builder

- 1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/ (https://console.aws.amazon.com/systems-manager/).
- 2. In the navigation pane, choose **Documents**.

-or-

If the AWS Systems Manager home page opens first, choose the menu icon () to open the navigation pane, and then choose **Documents** in the navigation pane.

- 3. Choose Create automation.
- 4. For Name, type this descriptive name for the runbook: LaunchInstanceAndCheckStatus.
- 5. (Optional) For **Document description**, replace the default text with a description for this runbook, using Markdown. The following is an example.

6. For **Assume role**, enter the ARN of the IAM service role for Automation (Assume role) for the automation, in the format arn:aws:iam::111122223333:role/AutomationServiceRole. Substitute your AWS account ID for 111122223333.

The role you specify is used to provide the permissions needed to start the automation.

▲ Important

For runbooks not owned by Amazon that use the aws:executeScript action, a role must be specified. For information, see Permissions for using runbooks (./automation-document-script.html#execution-permissions).

- 7. Expand Input parameters and do the following.
 - 1. For **Parameter name**, enter imageId.

latest/amzn-ami-hvm-x86 64-gp2 }}

- 2. For **Type**, choose String.
- 3. For **Required**, choose No.
- 4. For **Default value**, enter the following.

{{ ssm:/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2 }}

Note

This value launches an EC2 instance using the latest Amazon Linux Amazon Machine Image (AMI) ID. If you want to use a different AMI, replace the value with your AMI ID.

5. For **Description**, enter the following.

(Optional) The AMI ID to use for launching the instance. The default value uses the latest released Amazon Linux AMI ID.

- 8. Choose Add a parameter to create the second parameter, tagValue, and enter the following.
 - 1. For **Parameter name**, enter tagValue.
 - 2. For **Type**, choose String.
 - 3. For **Required**, choose No.
 - 4. For **Default value**, enter LaunchedBySsmAutomation. This adds the tag key-pair value Name: LaunchedBySsmAutomation to the instance.
 - 5. For **Description**, enter the following.

(Optional) The tag value to add to the instance. The default value is LaunchedBySsmAutomation.

- 9. Choose **Add a parameter** to create the third parameter, instanceType, and enter the following information.
 - 1. For **Parameter name**, enter instanceType.
 - 2. For **Type**, choose String.
 - 3. For **Required**, choose No.
 - 4. For **Default value**, enter t2.micro.
 - 5. For **Parameter description**, enter the following.

(Optional) The instance type to use for the instance. The default value is t2.micro.

- 10. Expand Target type and choose "/".
- 11. (Optional) Expand **Document tags** to apply resource tags to your runbook. For **Tag key**, enter Purpose, and for **Tag value**, enter LaunchInstanceAndCheckState.
- 12. In the **Step 1** section, complete the following steps.
 - 1. For **Step name**, enter this descriptive step name for the first step of the automation: LaunchEc2Instance.
 - 2. For **Action type**, choose **Run a script** (aws:executeScript).

3. For **Description**, enter a description for the automation step, such as the following.

```
**About This Step**

This step first launches an EC2 instance using the ```aws:executeScript``` action and the provided script.
```

- 4. Expand Inputs.
- 5. For **Runtime**, choose the runtime language to use to run the provided script.
- 6. For **Handler**, enter launch instance. This is the function name declared in the following script.
 - Note

This is not required for PowerShell.

7. For **Script**, replace the default contents with the following. Be sure to match the script with the corresponding runtime value.

```
Python
                    PowerShell
         (#python)
                                 (#powershell)
def launch instance(events, context):
  import boto3
  ec2 = boto3.client('ec2')
  image_id = events['image_id']
  tag_value = events['tag_value']
  instance_type = events['instance_type']
  tag_config = {'ResourceType': 'instance', 'Tags': [{'Key':'Name',
'Value':tag_value}]}
  res = ec2.run instances(ImageId=image id, InstanceType=instance type, MaxCount=1,
MinCount=1, TagSpecifications=[tag_config])
  instance id = res['Instances'][0]['InstanceId']
  print('[INFO] 1 EC2 instance is successfully launched', instance id)
  return { 'InstanceId' : instance_id }
```

- 8. Expand Additional inputs.
- 9. For Input name, choose InputPayload. For Input value, enter the following YAML data.

```
image_id: "{{ imageId }}"
tag_value: "{{ tagValue }}"
instance_type: "{{ instanceType }}"
```

- 13. Expand Outputs and do the following:
 - For Name, enter payload.
 - For **Selector**, enter \$.Payload.
 - For **Type**, choose StringMap.
- 14. Choose **Add step** to add a second step to the runbook. The second step queries the status of the instance launched in Step 1 and waits until the status returned is ok.
- 15. In the **Step 2** section, do the following.
 - 1. For **Step name**, enter this descriptive name for the second step of the automation: WaitForInstanceStatusOk.
 - 2. For **Action type**, choose **Run a script** (aws:executeScript).
 - 3. For **Description**, enter a description for the automation step, such as the following.

```
**About This Step**
```

The script continuously polls the instance status check value for the instance launched in Step 1 until the ```ok``` status is returned.

- 4. For **Runtime**, choose the runtime language to be used for executing the provided script.
- 5. For **Handler**, enter poll instance. This is the function name declared in the following script.
 - Note

This is not required for PowerShell.

6. For **Script**, replace the default contents with the following. Be sure to match the script with the corresponding runtime value.

```
Python (#python) PowerShell (#powershell)

def poll_instance(events, context):
   import boto3
   import time

ec2 = boto3.client('ec2')

instance_id = events['InstanceId']
```

```
print('[INFO] Waiting for instance status check to report ok', instance_id)
instance_status = "null"

while True:
    res = ec2.describe_instance_status(InstanceIds=[instance_id])

if len(res['InstanceStatuses']) == 0:
    print("Instance status information is not available yet")
    time.sleep(5)
    continue

instance_status = res['InstanceStatuses'][0]['InstanceStatus']['Status']

print('[INFO] Polling to get status of the instance', instance_status)

if instance_status == 'ok':
    break

time.sleep(10)

return {'Status': instance_status, 'InstanceId': instance_id}
```

- 7. Expand Additional inputs.
- 8. For Input name, choose InputPayload. For Input value, enter the following:

```
{{ LaunchEc2Instance.payload }}
```

16. Choose **Create automation** to save the runbook.

Step 2: Run the custom runbook

Use the following procedure to run the custom runbook created in Step 1. The custom runbook launches an EC2 instance and waits for the instance check to change to the ok status.

To run the custom runbook

- 1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/ (https://console.aws.amazon.com/systems-manager/).
- 2. In the navigation pane, choose **Automation**, and then choose **Execute automation**.
- 3. In the **Automation document** list, choose the **Owned by me** tab and then choose the button next to the custom runbook you created, LaunchInstanceAndCheckStatus.
- In the Document details section, for Document version, verify that Default version at runtime is selected.
- 5. Choose Next.

- 6. At the top of the **Execute automation document** page, verify that **Simple execution** is selected.
- 7. Choose Execute.
- 8. After both steps in the automation complete, in the **Executed steps** area, choose the step ID of a step to view steps details, including any step output.
 - Note

It can take several minutes for the ok status to be returned.

9. (Optional) Unless you plan to use the EC2 instance created by this walkthrough for other purposes, you can terminate the instance. For information, see Terminate Your Instance (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html) in the Amazon EC2 User Guide for Linux Instances.

You can identify the instance by the name **LaunchedBySsmAutomation** that you tagged it with in Step 1: Create the custom runbook (#automation-walk-document-builder-create) .