

# Step 1

[Click here](#) to Navigate to **CloudFormation > Stacks > Create stack (N. Virginia)**. Once you land up to the page, you need to put the below URL in the Amazon S3 URL as shown below & click Next. *Note: Make sure the Cloudformation Region and SSM Parameter Store Region are same, here we have used **N. Virginia Region**.*

**URL:** <https://automationking.s3.amazonaws.com/AnsibleParamStoreCFNTemplate.yml>

CloudFormation > Stacks > Create stack

Step 1  
**Specify template**

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

### Create stack

**Prerequisite - Prepare template**

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready  Use a sample template  Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL  Upload a template file

Amazon S3 URL  
  
Amazon S3 template URL

S3 URL: `https://automationking.s3.amazonaws.com/AnsibleParamStoreCFNTemplate.yml`

## Step 2

In this step first put a Stack name of your choice. Enter the below Input Parameters and click next.

1. In **AmazonLinux2ImageID** keep the default value, it picks the latest AMI ID of Amazon Linux 2 from the CloudFormation Region, as AMIs are the region-specific resource.
2. In **KeyName** select the Key-Pair which you will use to connect to the EC2 Instance (Ansible Controller Node), make sure you have the right Private Key in **pem format** physically stored in your Local System, as you need to import the Value in the next **section (1.2)**. Key-Pairs are region-specific resource so be careful and select the right Key-Pair. **If you don't have an EC2-Key Pair [Click here](#) to navigate to the Key-Pair Section (N.Virginia) & click Create Key-Pair in pem format.**
3. In **SSMGetParametersResource** the default value is \* which gives Ansible Controller Node to retrieve all Secure Strings from SSM Parameter Store, for simplicity you can keep the default value. To provide the least-privilege to the Ansible Controller Node, you can put the ARN of the SSM ParameteStore which you will create in the next section (1.2), refer the [AWS official document](#) for more details. Thus, to import the ARN of the ParameteStore in **SSMGetParametersResource**, you can start with section 1.2 and then move to section 1.1.
4. In **SourceSSHCIDRRange** the default value is 0.0.0.0/0 which opens the **Port 22** to the World, for simplicity you can keep the default value. Also, you can make the SSH connection restrictive by putting your ISP Public IP in CIDR Notation instead of 0.0.0.0/0.

CloudFormation > Stacks > Create stack

Step 1  
Specify template

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

### Specify stack details

**Stack name**

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**AmazonLinux2ImageID**  
Amazon Linux 2 Latest AMI ID

**KeyName**  
Select an existing EC2 key pair for SSH access to the EC2 instance. DISCLAIMER: MAKE SURE YOU HAVE THE EC2 KEY PAIR (PRIVATE KEY) WITH YOU.

**SSMGetParametersResource**  
Provide the ARN of the Parameter Store where you have stored the Private key. DEFAULT: \* (RETRIEVE ALL SECURE STRING PRESENT IN PARAMETER STORE) NOT RECOMMENDED.

**SourceSSHCIDRRange**  
Provide the Source CIDR Range to SSH to the AnsibleControllerNode. DEFAULT: 0.0.0.0/0 (FROM ANYWHERE) NOT RECOMMENDED

Cancel Previous Next

## Step 3

In this step just click next to proceed to Step 4.

CloudFormation > Stacks > Create stack

Step 1  
Specify template

Step 2  
Specify stack details

Step 3  
**Configure stack options**

Step 4  
Review

### Configure stack options

#### Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

<input type="text" value="Key"/>	<input type="text" value="Value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add tag"/>		

#### Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name ▾	<input type="text" value="Sample-role-name"/>	<input type="button" value="Remove"/>
-----------------	---	---------------------------------------

#### Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

- ▶ **Stack policy**  
Defines the resources that you want to protect from unintentional updates during a stack update.
- ▶ **Rollback configuration**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)
- ▶ **Notification options**
- ▶ **Stack creation options**

Cancel

## Step 4

In this step, scroll down to the end and check-mark **I acknowledge that AWS CloudFormation might create IAM resources** & click Create Stack to start the Ansible Environment build.

### Notification options

No notification options  
There are no notification options defined

### Stack creation options

Rollback on failure  
Enabled

Timeout  
-

Termination protection  
Disabled

► Quick-create link

### Capabilities

**ⓘ The following resource(s) require capabilities: [AWS::IAM::Role]**  
This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources.

Cancel Previous Create change set **Create stack**

## Step 5

[Click here](#) to Navigate to **EC2 Instance Dashboard (N. Virginia)**. Once you land up to the page, Look for the Running Instances; *AnsibleControllerNode* and *AnsibleTragetNode*. Note down the **Public Address of AnsibleControllerNode** to SSH to that Node & run playbooks and **Private IP of AnsibleTragetNode** which will get updated in the Playbook Host File.

The screenshot shows the AWS Management Console interface for the EC2 Instance Dashboard. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below this is a search bar with two filters: 'search: AnsibleControllerNode' and 'search: AnsibleTragetNode'. A table lists two instances:

Name	Environment	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public IP
AnsibleControllerNode		i-0b003c1e219548fc1	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-100-27-13-117
AnsibleTragetNode		i-0dbb3a23d57d20df4	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-16-165-44-242

The 'AnsibleControllerNode' instance is selected, and its details are shown below. The 'Description' tab is active, displaying various instance attributes:

- Instance ID: i-0b003c1e219548fc1
- Instance state: running
- Instance type: t2.micro
- Private DNS: ip-10-10-10-86.ec2.internal
- Private IPs: 10.10.10.86
- VPC ID: vpc-076f8c8aeeb3967b0
- Platform: Amazon Linux
- Platform details: Linux/UNIX
- Public DNS (IPv4): ec2-100-27-13-117.compute-1.amazonaws.com
- IPv4 Public IP: 100.27.13.117
- Availability zone: us-east-1e
- Security groups: AnsibleStack-AnsibleGenericSG-14660NUF2VZI3
- AMI ID: amzn2-ami-hvm-2.0.20201126.0-x86\_64-gp2
- Subnet ID: subnet-0fc08323e8b7f018b
- Network interfaces: eth0

The screenshot shows the AWS Management Console interface for the EC2 Instance Dashboard, similar to the previous one. The search filters are the same. The table lists the instances, and 'AnsibleTragetNode' is now selected:

Name	Environment	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public IP
AnsibleControllerNode		i-0b003c1e219548fc1	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-100-27-13-117
AnsibleTragetNode		i-0dbb3a23d57d20df4	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-16-165-44-242

The details for 'AnsibleTragetNode' are shown below. The 'Description' tab is active, displaying various instance attributes:

- Instance ID: i-0dbb3a23d57d20df4
- Instance state: running
- Instance type: t2.micro
- Private DNS: ip-10-10-10-28.ec2.internal
- Private IPs: 10.10.10.28
- VPC ID: vpc-076f8c8aeeb3967b0
- Platform: Amazon Linux
- Public DNS (IPv4): ec2-54-165-44-242.compute-1.amazonaws.com
- IPv4 Public IP: 54.165.44.242
- Availability zone: us-east-1e
- Security groups: AnsibleStack-AnsibleGenericSG-14660NUF2VZI3
- AMI ID: amzn2-ami-hvm-2.0.20201126.0-x86\_64-gp2
- Subnet ID: subnet-0fc08323e8b7f018b
- Network interfaces: eth0