When working with AWS CloudFormation templates, I often wish I had access to a richer set of parameter configuration options. For example, conditional parameters or even checkboxes would be a huge boost for user experience when selecting parameters for complex CloudFromation templates. Although some of these features can be hacked into a template, they often result in an overly clunky set up process, or complex (over)use of lambda backed custom resources.

One of the alternatives I have been working with recently is essentially breaking a CloudFormation template into two parts. The first template generates a static signed s3 backed website, where we can leverage the front end web stack (HTML, CSS, JavaScript) to create a custom set up page for our "actual" CloudFormation stack. Based off the results of this set up page, we dynamically construct the second template, which is really just a new custom CloudFormation template. Once the user has selected the options for the template, they can download the template and deploy it as they would any other CloudFormation Template – through either the AWS CLI or the console.

Unfortunately, the set up page and actual process of dynamically constructing a template is not particularly standardizable due to the massive variety of template use cases this approach can support, however, the process of creating a static signed s3 backed website from a CloudFormation template is repeatable and something you can mostly copy paste (There is a link to an example below). Based on your requirements, you will need to modify the HTML, CSS, and JavaScript of the static s3 site to provide all of the settings and options you would like your template to support, and of course you will need to write the JavaScript that dynamically creates the template based on the user selections. Keep in mind that as CloudFormation templates can be in JSON, it makes things easier to treat the template as a big JSON artifact within your JavaScript code, which makes modifications and additions easier.

It's not the most straightforward process, however it gives you enormous flexibility over the setup process of a CloudFormation template, and for complex parameter options, is far more user friendly than trying to use the CloudFormation setup parameter primitives.

In this example (github link) I used this process to create a set up page where a user selects from aa list of AWS service logs that they would like to forward to Splunk using CloudWatch Metrics, CloudWatch Events and Kinesis Streams/Firehose. You can see that the set up page really just contains a list of checkboxes and a few textboxes that depend on previous parameter selections, however these setup primitives are currently unavailable in CloudFormation, and as I have full control over the UI of the set up page, I could easily add more complex set up options, longer descriptions, embedded videos, etc. You'll probably notice that I have also taken the liberty of making the setup UI a little bit more "Splunk-y".

Once I finish the setup process, I download the custom generated template, and launch the stack using the AWS CLI or AWS CloudFormation console.

Hopefully this was helpful! It may be too complex a workaround for some use cases, but for our purposes, it made for a much smoother and more customizable CloudFormation stack deployment process - despite the added intermediary step.