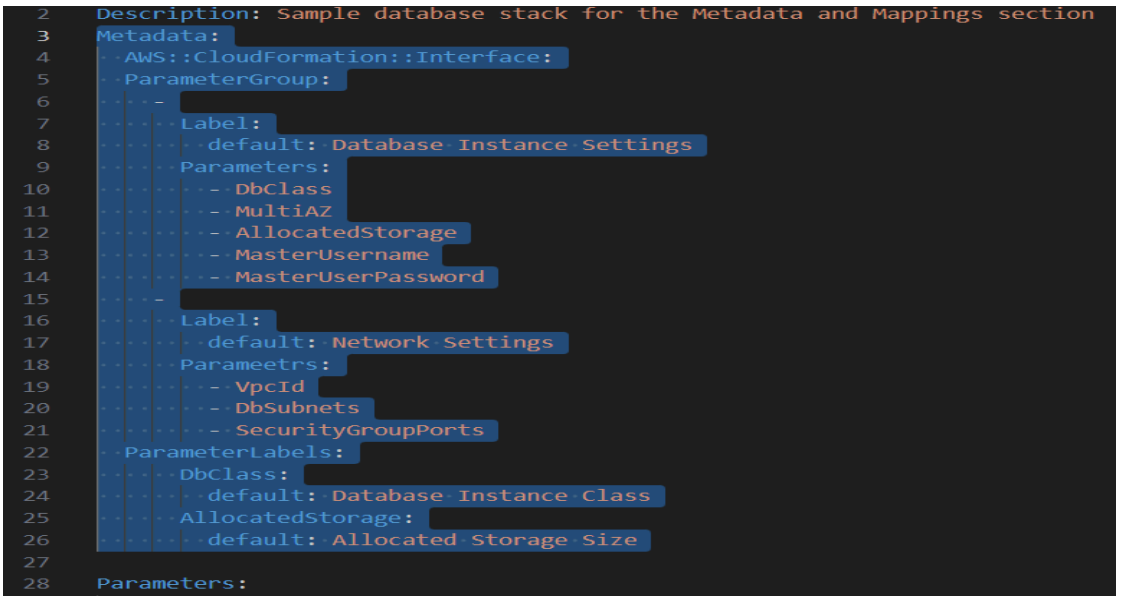
Grouping, Ordering and Labelling Parameters

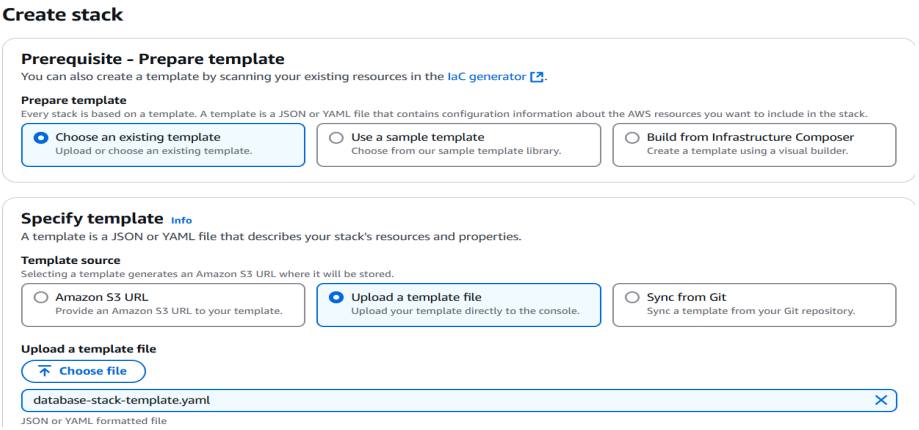
The process involves enhancing a CloudFormation template by adding a Metadata section for grouping, ordering, and customizing parameter labels. Use AWS::CloudFormation::Interface to define ParameterGroups and specify parameter order and display labels. Update ParameterLabels to assign custom display names. Save the edited template, upload it to AWS CloudFormation, and create a stack with a unique name (e.g., DatabaseStack). Provide parameter values and submit the stack, observing grouped and labeled parameters in the creation console. Verify the stack's successful creation to confirm the template's functionality. Finally, delete the stack to clean up resources and avoid unnecessary costs.

Activity

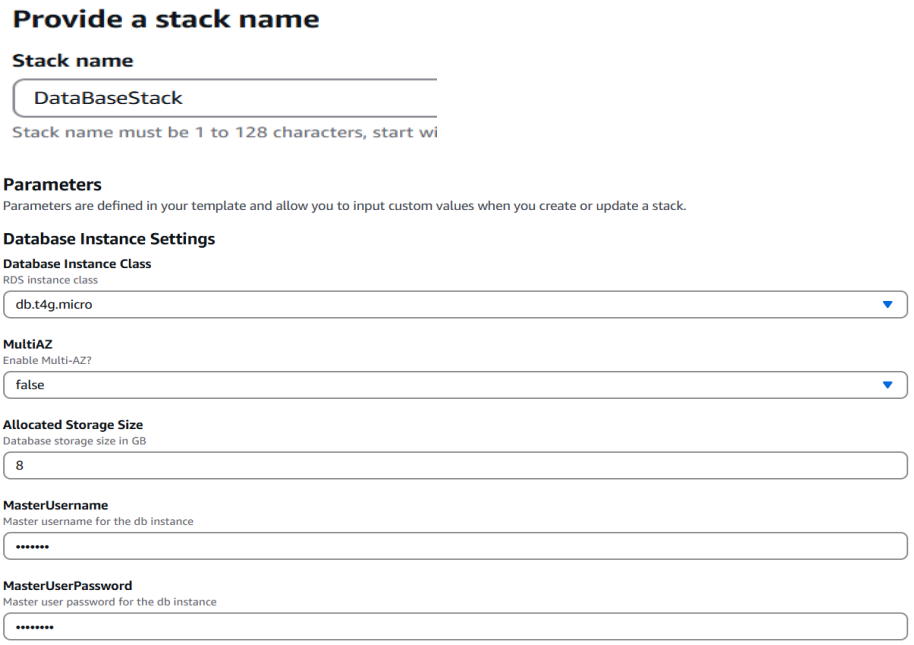
1. Find the template files in our GitHub repository under the same name as the heading for easy access and edits. Find and Save the attached template locally, open it in VS Code for edits.
2. Review the template and Add the Metadata section at the top of the template before the Parameters section. Defined AWS::CloudFormation::Interface as the metadata key for grouping and ordering parameters. Created a ParameterGroups key under metadata and listed groups in the desired order. Added a Label attribute to each group with a Default value for its display name. Specified group members in a Parameters key under each group, listing parameter names. Defined ParameterLabels under metadata to customize parameter display labels. Customized labels for specific parameters using key-value pairs and Default attributes.



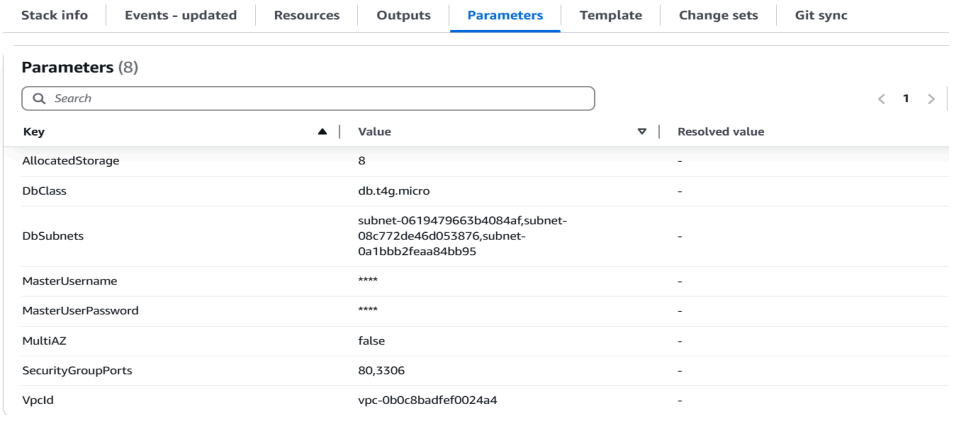
3. Change the Image Id, Subnet id and Vpc id according to your region.Save the updated template.
4. Upload the template to CloudFormation and create a stack.



5. Provide a unique stack name e.g., '**DataBaseStack**' and fill in the parameters as shown. Click on '**NEXT**' to continue on the review page and submit.



6. Observed grouped, ordered, and custom-labeled parameters on the stack creation console.



7. Verified the functionality and correctness of the template by successfully creating the stack.
8. Deleted the stack to clean up resources.