**Elex ARM Templates - Steps For Deploy Resources**

Get Templates:

1. Templates are located in a public GitHub repository with a corresponding parameter file.

* Repository Link: <https://github.com/Elexratio-ARM/Elex-ARM-Templates>
* The templates are categorized based on resource names and also include a comprehensive template for setting up the entire environment.

1. Download all the templates and parameter files from the GitHub repository.
2. Upload the downloaded files to your Azure repository, placing them in the appropriate folder.

A screenshot of a computer

Description automatically generated

Templates are available for the following resources:

* Environment Setup Template(Contain All Resources)
* App Service With Config Setting,
* Key Vault with Empty Secrete keys,
* Storage Account with containers,
* SQL Database.

A screenshot of a computer

Description automatically generated with medium confidence

To upload all the template files to Azure Repos and organize them by resource name like Git Repository.

**Set up Release Pipeline:**

The next step is to set up the release pipeline to deploy the ARM template. This pipeline will deploy all the resources required to set up a new environment.

1. Open the Azure DevOps portal that is linked with Azure AD.
2. Go to the "Releases" section in the Pipeline side menu.
3. Click on the "New" button located in the corner.
4. Click New Release Pipeline in the Listed options.

A screenshot of a computer

Description automatically generated

* Next step is to set Empty job in the release pipeline

A screenshot of a computer

Description automatically generated with medium confidence

* Then click the add an artifacts in artifacts section

A screenshot of a computer

Description automatically generated

Configure the Resource Type of the Release Pipeline:

* The next step is to configure the resource type of our release pipeline. This involves selecting the repository and branch for the template file that has been committed.
* Click the "Add" button after filling in all the required sources.

A screenshot of a computer

Description automatically generated

Next step is setup task for release pipeline.

1. In the Tasks section, click on the plus icon within the agent job.
2. Use the search box to search for "ARM". It will display the "ARM Template deployment" job below.
3. Add that job to the task and then we can configure it .

A screenshot of a computer

Description automatically generated with medium confidence

**Configuring Environment Deploy Setup**:

A screenshot of a computer

Description automatically generated with medium confidence

1. Fill in the Azure mandatory details in the configuration settings.
2. The first one is the Deployment Scope, set it as "Resource Group".
3. Configure the Azure Resource Manager connection. This connection is used for connect to the Azure portal. Select the appropriate connection for your Azure Portal.
4. The subscription for your Azure portal will be show automatically when you configure the Azure Resource Manager connection.
5. Set the Action as "Create or update Resource Group".
6. Now, select the Resource Group where you want to deploy all the resources.

A screenshot of a computer

Description automatically generated with medium confidence

1. Select the location for the Resource Group where the resources will be deployed.
2. Start by deploying the Environment Setup Template. This template deploys the resources necessary to set up the new environment.
3. Choose the location of the Environment Deploy Template file. Since the files were committed directly, select "Linked artifact".
4. Select the parameter file for the Environment Deploy Template resource. If needed, click on "Override Template Parameter" to override the naming for all resources.
5. Select Deployment Mode as "Incremental".

A screenshot of a computer

Description automatically generated

Here, we have the option to change the resource name, by selecting override template parametes file but make sure to set it according to the naming convention for the resources.

1. The first step is to deploy the App Service, which requires an App Service plan to host the service. The template will check whether the provided App Service plan name already exists or if it needs to create a new one.
2. If the provided App Service plan name already exists, the App Service will be created under that existing plan.
3. If the App Service plan is new, the template will create a new App Service plan in the provided appServicePlanResourceGroup first, and then deploy the App Service under that newly created plan.
4. The SQL Server name is a prerequisite for creating the SQL Database. The template will check if the provided SQL Server name already exists. If it does not exist, the template will create a new SQL Server with the provided admin username and password for the specified location.

This ensures that the SQL Server is available for the SQL Database deployment. If the SQL Server already exists, the template will utilize the existing SQL Server for the database deployment.

Once all the values are filled, click on the "Save and Release" button.

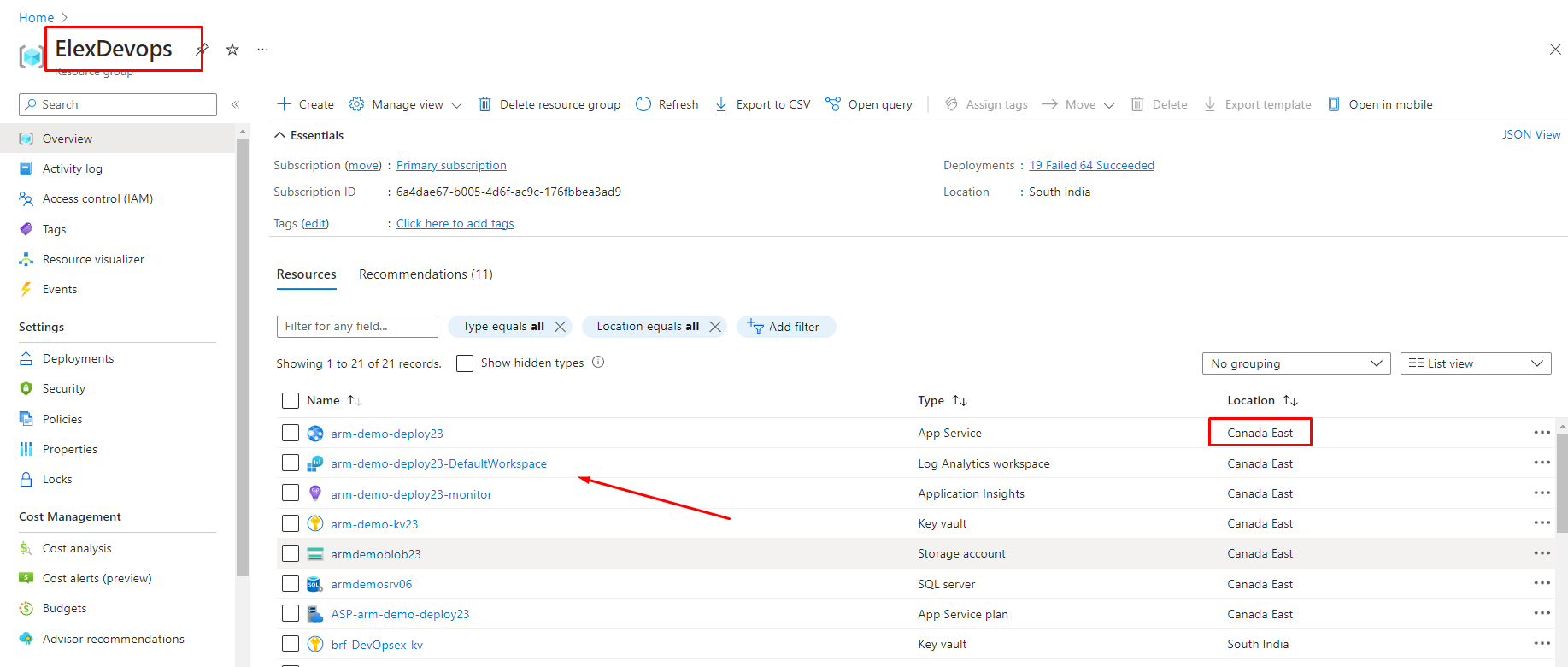
A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

When the job succeeds, all the resources which need for new Environment will be created in the Azure portal.



**Configuring Separate Resource setup:**

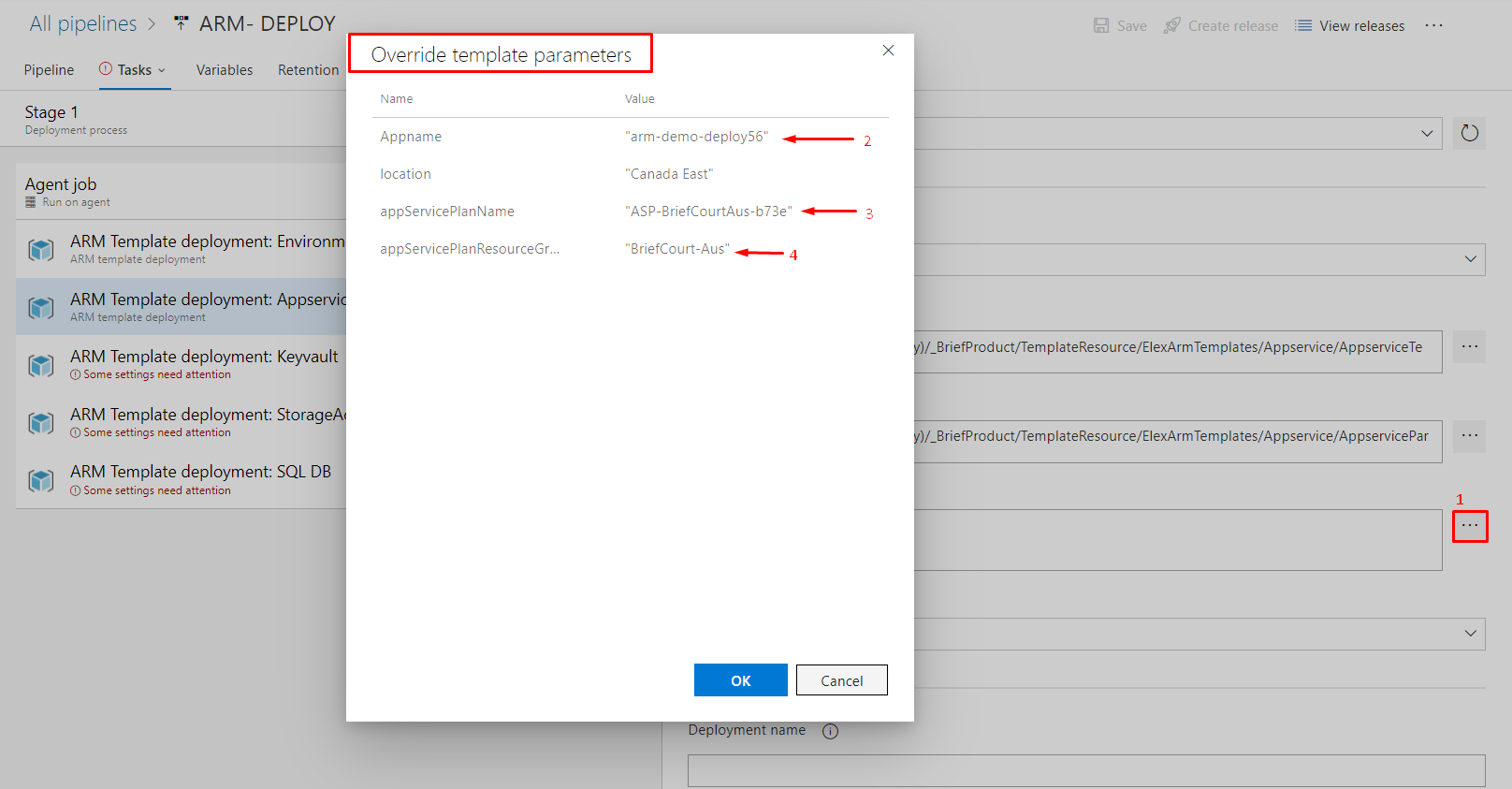
If you want to deploy individual resources like App Service, Key Vault, or Blob Storage separately, you can still utilize template files specifically designed for each resource. Instead of deploying the entire environment setup, you can choose to deploy only the necessary resources**.**

Here are the steps to deploy a solo resource using its template file

A screenshot of a computer

Description automatically generated with medium confidence

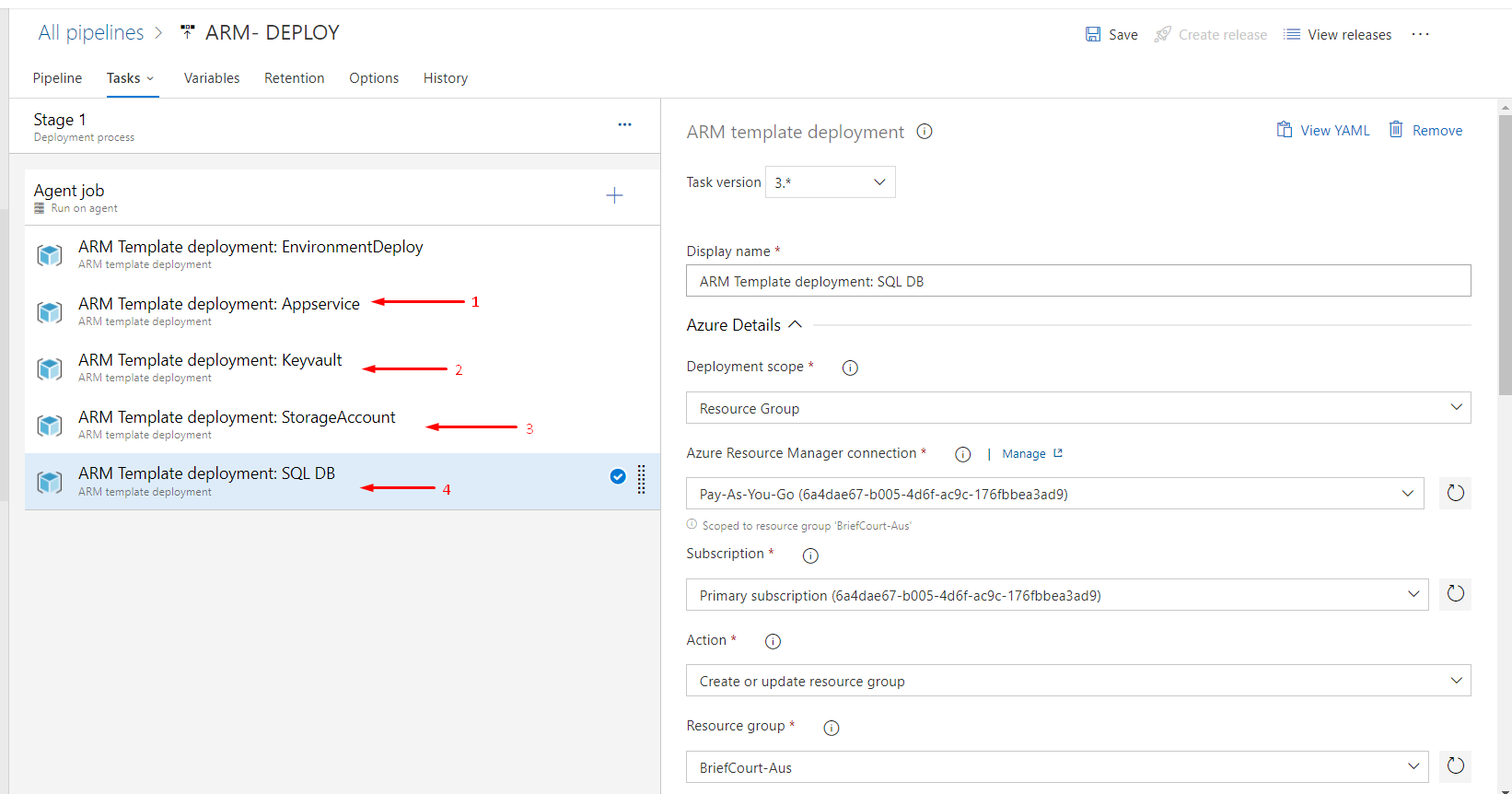
Select the App Service template file and parameter file which we commit the repository earlier in the specific folder [Appservice]



* Choose the override template parameter file, where you can configure the App Name, App Service Plan and Resource Group that contains the App Service Plan.

Once all the configurations are completed, click the "Save and Create Release" button. This will initiate the release process and deploy the App Service to the designated resource group.

Similarly Following these steps We can deploy the other resources.



* In a similar manner, you can deploy Key Vault, Storage Account, and SQL Database using their respective template files. Additionally, you have the option to disable the job if you don't want to create the Resources like Blob or SQL Database.

In conclusion, by following the aforementioned steps, you can successfully deploy ARM templates and create resources in the Azure portal.