1. **What is a Configuration?**

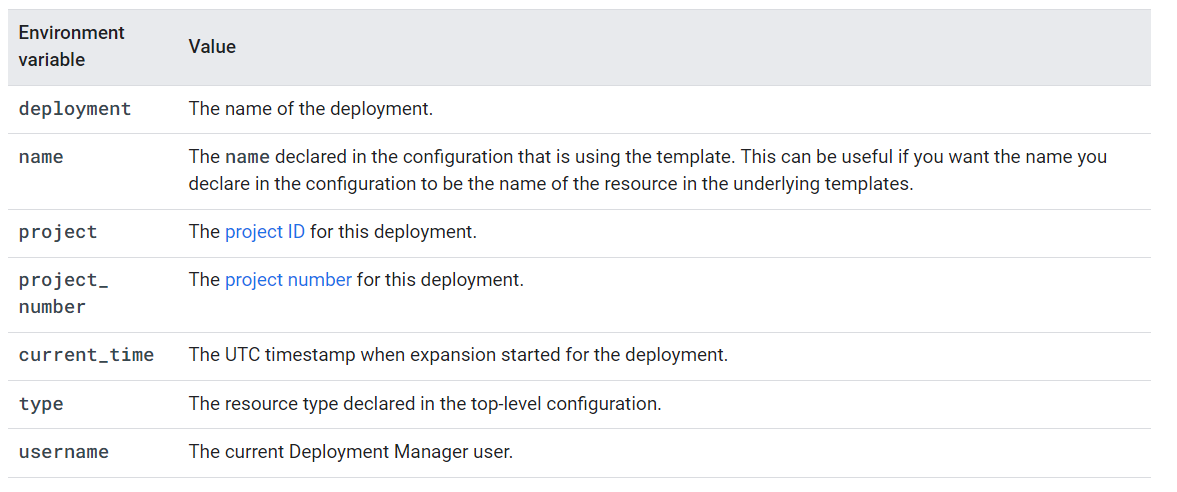
* A Configuration defines the structure of your deployment.
* A Configuration describe all the resources you want for a single deployment and is written in yaml syntax.
* A Configuration must contain resources section (list if resource to create)
* Deployment Manager can be a powerful tool in your tool belt when it comes to implementing **Infrastructure as a code (IaaC - IaC)**

1. **Components**

* Each resource has 3 Components:
  + Name: is user-defined string to identify resource (ex: instance-1)
  + Type: used to create part of deployment, include:
    - Base Type: represents a single of api source
      * Syntax: [API].[VERSION].[RESOURCE]
    - Composite Type: set of resources, contains one or more templates that are pre-configured to work together. These templates expand to set of base type. Essentially hosted templates.
      * Syntax: gcp-types/[PROVIDER]:[RESOURCE] (ex: gcp-types/compute-v1:address)
  + Properties: is the parameters for the resource type, it includes these parameters: zone, machineType, boot (disk type), sourceImage. The paremeters must be match those of resources.

1. **Templates**

* A Template is essentially a part of the configuration file that have been abstracted into individual building block
* A Template is a separate file that is imported and used as a type in a Configuration
* A Configuration can contain many Templates, you can use many templates as you want in a Configuration
* It allows you to separate your Configuration out into different pieces that you can use and resuse across different deployments.
* It allows you to take advantage of features like templates properties, environment variables and modules to create dynamic Configuration
* Advantages:
  + Template properties: are arbitrary variables that you define in template files. Any Configuration or Template can provide a value for the template property without changing the template directly (abstract the property so that you can change property value in each Configuration).
    - Deployment Manager creates predefined environment variables that you can use in your deployment.
* Available environment available:





**Schemas of Template:**

* Schema describe the specifications of a Deployment Manager template, it defines a set of rules that a configuration file must meet if it wants to use a particular template.
* In addition to defining the rules of a template, schemas also allow your users to interface with the templates you write, without needing to review and learn about each layer of templates.
* User can simply review the requirements defined in your schema instead to learn what properties are settable or required for the perspective template.

1. **Deployment**

* Deployment is a collection of resources that are deployed and managed together using a Configuration
* We can deploy, update or delete the Deployment by merely changing some code or at a click of a button
* A Deployment can contain number of resources across number of google cloud services
* Each Deployment has a corresponding **Manifest**, a **Manifest** is a read-only property that describes all the resources in your deployment, and is automatically created with each new deployment.
* **Manifest** can not be modified after they have been created, it is created based on the Configuration file.

1. **Deploy**

* When you deploy, you need to provide a valid Configuration in the request to create a Deployment.
* When you deploy a deployment, Deployment Manager creates all of the described resources to deploy a Configuration, it must be done through the command line:
  + Command line to deploy a deployment: ***gcloud deployment-manager deployments create bowtie-deploy --config bowtie-deploy.yaml --preview***

1. **Update**

* We can add or remove any resources from a Deployment, or update the properties of existing resources in a Deployment.
* We can both adding a new resource and making changes to properties of existing resources in the same request.
* To update a Deployment, we can update the Configuration file or create a Configuration file with the changes.
* Each update will have a policy (either picked policies or default policies)
* Step: Update Configuration file -> Choose policies -> Make the update request to deployment manager.

1. **Delete**

* When you delete a deployment, all resources that are part of the deployment are also deleted
* If you want to delete specific resources from your Deployment and keep the rest, delete those resources from your Configuration file and update the deployment

1. **Best Practices**

* **Break up your Configuration into logical units:** we should create separate Configurations for networking services, security services and compute services.
* **Use references:** enforces order resources are created, should be used for values that are not defined until a resource is created such as **resources self-link**, **ip address**, **system generated id**
* **Preview your deployment:** using the preview flag, you should always preview your deployment to access how making an update will affect your deployment. This option is to see the changes before committing to it
* **Automate the creation of resources:** consider to automate the creation of project and automate the creation of resources contained within the projects. This enables you to adopt an **infrastructure as a code (IaaC)** approach for project provisioning, this will allow you to provide a series of predefined project environments that can be quickly and easily provisioned
* **Use version control:** using a version control system as part of the development process for your deployments is a great best practice to follow as it allows you to fall back:
  + Previous known good config
  + Audit trail
  + Use Config for CI/CD