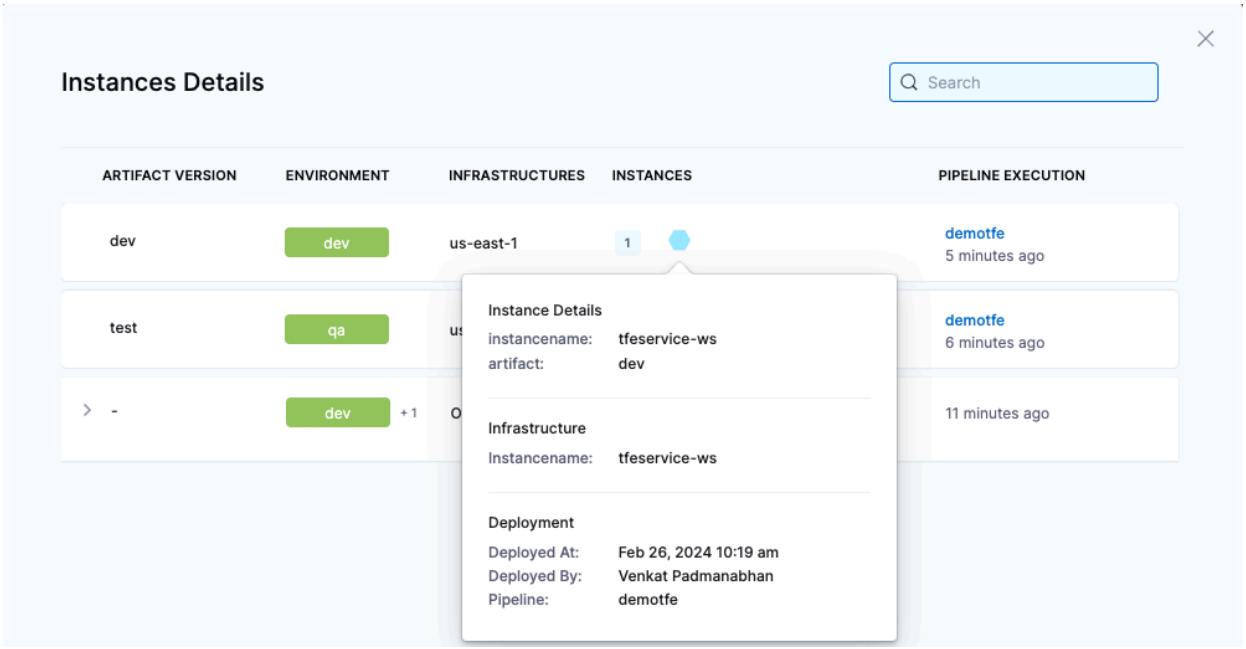
# TFE DEPLOYMENT TEMPLATE

## OVERVIEW

Using deployment template allows users to design a pipeline that is similar to the out of the box capabilities available for other workloads such as kubernetes deployment.

Deployment template based pipeline allows users to leverage Harness platform capabilities such as services, environment and security that allows to track and view the execution details but also configure RBAC for additional security.




## HIGH LEVEL DESIGN

Click the component link to get the yaml for the corresponding entity. The yaml for infra definition is in the description.

| Component | Description |
| --- | --- |
| Deployment Template | Common foundation for all components.<br><br>Defines the Fetch instance script, input and mapping of the json payload to instance. |
| Stage Template | Created using Deployment Template. Contains execution steps |
| Service | Created using Deployment Template.<br>Configures artifact source, config files and variables |
| Infra Definition | Created using Deployment Template.<br>Configures infrastructure properties and how the values are provided.<br>dev-infra<br>qa-infra |
| Pipeline | Uses the stage template to create one or more stages.<br>Configures pipeline variable and defines fixed values for stages to limit the user input during the execution |

## FILESTORE

Use filestore to create files with harness expressions. Notice the use of config files and filestore. Config files are associated with service and can be configured with overrides. If the contents of the file are not different based on the environment or service i.e there is no need for using overrides, then replace the use of config files with filestore.

These files are referenced in shell script as follows:

```
cat <<_EOF_ > /workspace/credentials.tfrc.json
<+configFile.getAsString("credentials")>
_EOF_
```

```
  cat <<_EOF_ > /workspace/_override.tf
  <+fileStore.getAsString("/TFE/override.tf")>
  _EOF_
```

**/TFE/credentials.tfrc.json**

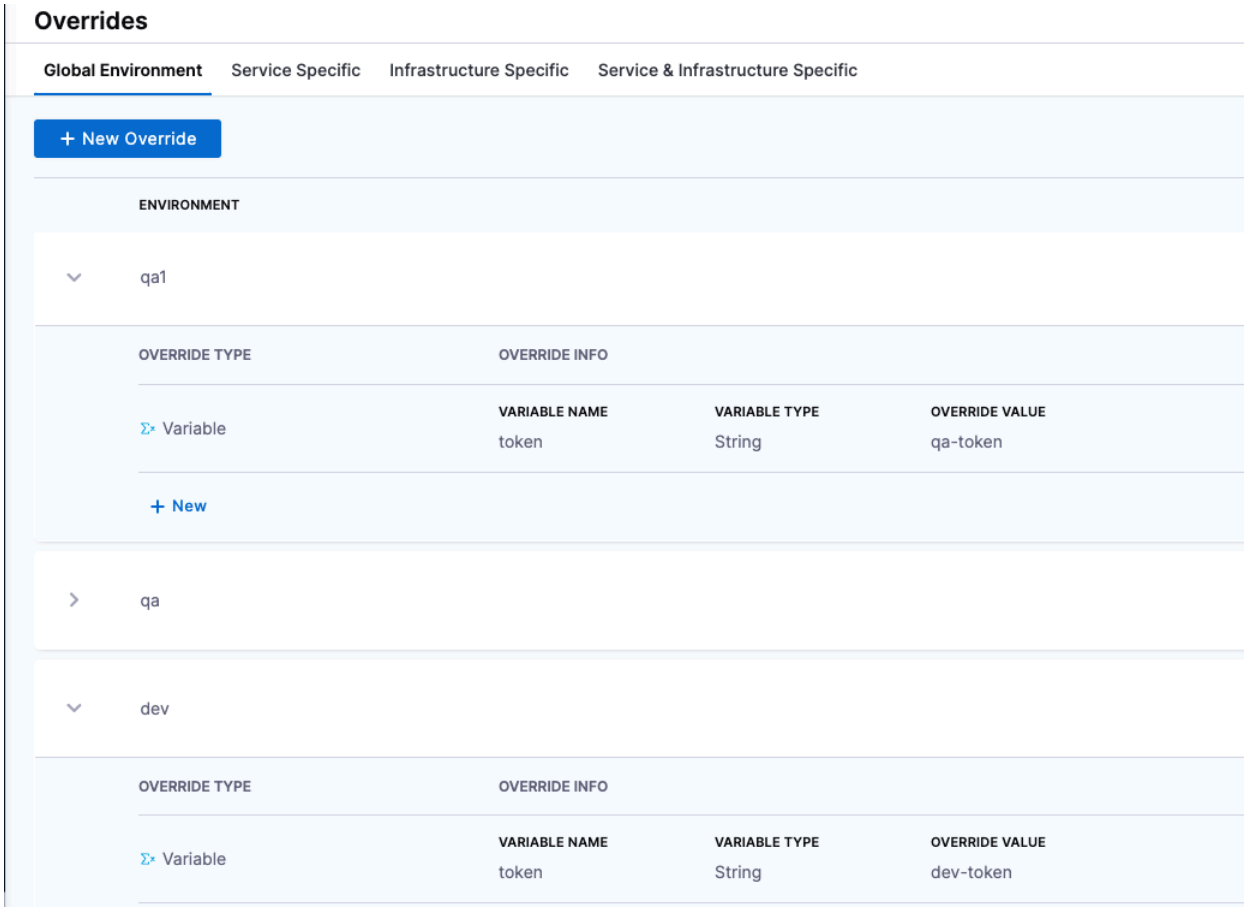
```
{
 "credentials": {
   "app.terraform.io": {
     "token": "<+secrets.getValue("demouserkey")>"
   }
 }
}
```

**/TFE/override.tf**

```
terraform {
 backend "remote" {
   hostname = "app.terraform.io"
   organization = "<+infra.variables.ORG>"
   workspaces {
     name = "<+infra.variables.WORKSPACE>"
   }
 }
}
```

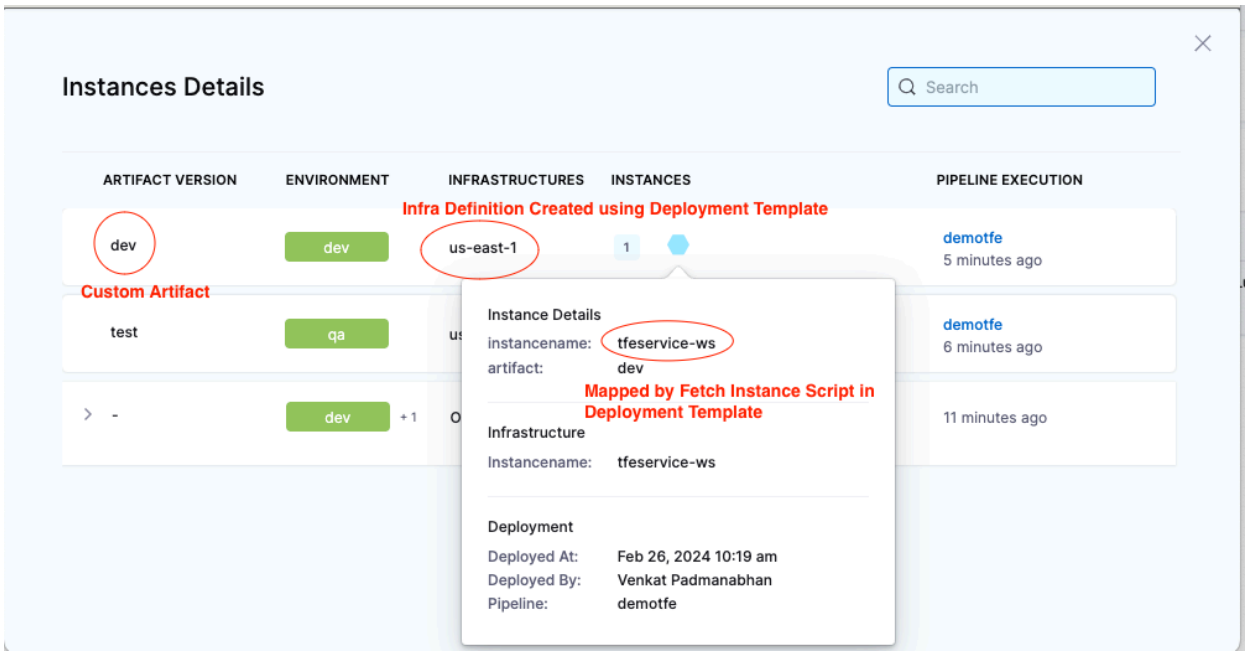
## OVERRIDES

Overrides enable pre-defined and consistent fixed input based on environment to eliminate unnecessary user input and reducing manual error.




## Reference

1. Deployment Template
2. Overrides


Instances Details
Search
ARTIFACT VERSION
ENVIRONMENT
INFRASTRUCTURES
INSTANCES
PIPELINE EXECUTION
Infra Definition Created using Deployment Template
dev
dev
us-east-1
1
demotfe
5 minutes ago
Custom Artifact
test
qa
demotfe
6 minutes ago
-
dev
+1
11 minutes ago
Instance Details
instancename: tfeservice-ws
artifact: dev
Mapped by Fetch Instance Script in Deployment Template
Infrastructure
Instancename: tfeservice-ws
Deployment
Deployed At: Feb 26, 2024 10:19 am
Deployed By: Venkat Padmanabhan
Pipeline: demotfe