# Cloud Deployment JSON Structure

```
{
  "application": "flowtest",    //application name
  "id": "6c8204d0-0672-4a44-b1ee-016d6e543194", //id
  "index": 3,                                   //ignore
  "keepWaitingPipelines": false,
  "lastModifiedBy": "anonymous",
  "limitConcurrent": true,
  "name": "test3",     //pipeline_name
  "stages": [],   //Stages are defined down.
"triggers": [],
  "parameterConfig": [],
  "notifications": [],
  "description": "descriptiom"
```

# Stages Information

### 1->Deploy Image

```
"stages": [
   {
     "clusters": [
       {
         "account": "my-k8s-account",
         "application": "flowtest",
         "cloudProvider": "kubernetes",
         "containers": [ //Select at least one image to run in this server group (pod). If multiple images are selected, they will be colocated and replicated equally
```

```
 {
  "args": [],
  "command": [],
  "envFrom": [],
  "envVars": [],
  "imageDescription": {
   "imageId": "index.docker.io/vishnuag1/spring-server:1.0.8", //The image selected under Basic Settings whose container is to be configured.
   "registry": "index.docker.io", //The registry the selected image will be pulled from.
   "repository": "vishnuag1/spring-server",
   "tag": "1.0.8"
  },
  "imagePullPolicy": "IFNOTPRESENT",  //Sets the policy used to determine when to pull (download) the selected container image.



  "name": "vishnuag1-spring-server",
  "ports": [
   {
    "containerPort": 8080, //The port to expose on this container.
    "hostIp": "", //The IP to bind the external port to. Most containers do not need this.
    "hostPort": 8074, //The port to expose on Host IP. Most containers do not need this
    "name": "http", //A name for this port. Can be found using DNS lookup if specified.
    "protocol": "TCP //The protocol for this port
   }
  ],
  "volumeMounts": []
 }
],
```

```
"deployment": {
  "enabled": false
},
"dnsPolicy": "ClusterFirst", //Set DNS policy for containers within the pod.
"events": [],
"initContainers": [],
"interestingHealthProviderNames": [
  "KubernetesContainer",
  "KubernetesPod"
],
"loadBalancers": [
  "flowtest-ft-ft1"
],
"namespace": "default",
"nodeSelector": {},
"podAnnotations": {},
"provider": "kubernetes",
"region": "default",
"replicaSetAnnotations": {
  "service.spinnaker.io/enabled": "false"
},
"rollback": {
  "onFailure": true
},
"scaleDown": false,
"securityGroups": [],
"stack": "baseline", //One of the core naming components of a cluster, used to create vertical stacks of dependent services for integration testing.
```

```
        "strategy": "redblack", //The deployment strategy tells Spinnaker what to do with the previous version of the server group.
        "targetSize": 1,
        "terminationGracePeriodSeconds": 30,
        "volumeSources": []
      }
    ],
    "comments": "j",
    "name": "Deploy",
    "notifications": [
     {
      "address": "vishnu.agarwal@infosys.com",
      "level": "stage",
      "type": "email",
      "when": [
       "stage.starting",
       "stage.complete"
      ]
     }
    ],
    "refId": "3",
    "requisiteStageRefIds": [],
    "sendNotifications": true,
    "type": "deploy"
  },
```

## 2->Add PreConditions *//Checks for preconditions before continuing*

```
{
```

```
    "name": "Check Preconditions",
    "preconditions": [
     {
      "cloudProvider": "kubernetes",
      "context": {
       "cluster": "flowtest-baseline", //The cluster to look at when selecting the image to use in this pipeline
       "comparison": "<", //comparsison for expected
       "credentials": "my-k8s-account",
       "expected": 3, //Number of server groups in the selected cluster
       "moniker": {
        "app": "flowtest",
        "cluster": "flowtest-baseline",
        "stack": "baseline"
       },
       "regions": [
        "default"
       ]
      },
      "failPipeline": true, //the overall pipeline will fail whenever this precondition is false
      "type": "clusterSize"
     }
    ],
    "refId": "4",
    "requisiteStageRefIds": [
     "3"
    ],
    "type": "checkPreconditions"
   }
```

## 3->Destroy Server Group *//Destroy Server Group Destroys a server group*

```
{
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline", //The cluster to look at when selecting the image to use in this pipeline
    "credentials": "my-k8s-account",
    "interestingHealthProviderNames": [
      "KubernetesService"
    ],
    "name": "Destroy Server Group",
    "namespaces": [
      "default"
    ],
    "refId": "5",
    "requisiteStageRefIds": [
      "4"
    ],
    "target": "current_asg_dynamic",
    "type": "destroyServerGroup"
  },
  {
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline",
    "credentials": "my-k8s-account",
    "interestingHealthProviderNames": [
```

```
      "KubernetesService"
    ],
    "name": "Disable Cluster",
    "namespaces": [
      "default"
    ],
    "preferLargerOverNewer": "false",
    "refId": "6",
    "remainingEnabledServerGroups": 1,
    "requisiteStageRefIds": [
      "5"
    ],
    "type": "disableCluster"
  }
```

## 4->Disable Server Group *//Disable Server Group Disables a server group*

```
{
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline", //The cluster to look at when selecting the image to use in this pipeline
    "credentials": "my-k8s-account",
    "interestingHealthProviderNames": [
      "KubernetesService"
    ],
    "name": "Disable Server Group",
    "namespaces": [
      "default"
```

```
    ],
    "refId": "7",
    "requisiteStageRefIds": [
      "6"
    ],
    "target": "current_asg_dynamic",
    "type": "disableServerGroup"
  }
```

## 5->Enable Server Group *// Enable Server Group Enables a server group*

```
{
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline", //The cluster to look at when selecting the image to use in this pipeline
    "credentials": "my-k8s-account",
    "interestingHealthProviderNames": [
      "KubernetesService"
    ],
    "name": "Enable Server Group",
    "namespaces": [
      "default"
    ],
    "refId": "8",
    "requisiteStageRefIds": [
      "7"
    ],
    "target": "current_asg_dynamic",
```

```
    "type": "enableServerGroup"
  }
```

## 6-> Resize Server Group *//Resize Server Group Resizes a server group*

```
{
    "action": "scale_up", //Configures the resize action for the target server group
    "capacity": {},
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline",
    "credentials": "my-k8s-account",
    "failPipeline": true,
    "judgmentInputs": [],
    "name": "Resize Server Group",
    "namespaces": [
      "default"
    ],
    "notifications": [],
    "refId": "9",
    "requisiteStageRefIds": [
      "8"
    ],
    "resizeType": "incr",
    "scaleNum": 34,
    "target": "current_asg_dynamic", //Select the deployed server when this pipeline starts
    "type": "resizeServerGroup"
  }
```

## 7->Scale Down Cluster *// Scale Down Cluster Scales down a cluster*

```
{
    "allowScaleDownActive": false,
    "application": "flowtest",
    "cloudProvider": "kubernetes",
    "cloudProviderType": "kubernetes",
    "cluster": "flowtest-baseline",
    "credentials": "my-k8s-account",
    "name": "Scale Down Cluster",
    "namespaces": [
      "default"
    ],
    "preferLargerOverNewer": "false",
    "refId": "10",
    "remainingFullSizeServerGroups": 1, //The remaining server groups will be scaled down to zero instances.
    "requisiteStageRefIds": [
      "9"
    ],
    "type": "scaleDownCluster"
  }
```

## 8-> Script *//Script Runs a script*

```
{
    "account": "",
    "cluster": "",
```

"cmc": "", *//(Optional) cmc passed down to script execution as CMC*

"command": "", *//Executable script and parameters. (e.g. script.py --ami-id ${deploymentDetails[0].ami} )*

"failPipeline": true,

"image": "", *//(Optional) image passed down to script execution as IMAGE_ID/*

"name": "Script",

"propertyFile": "", *//(Optional) The name to the properties file produced by the script execution to be used by later stages of the Spinnaker pipeline.*

"refId": "11",

"region": "", *//(Optional) region passed down to script execution as REGION_PARAM*

"repoBranch": "", *//Git Branch. (e.g. master). Leave empty to use the master branch.*

"repoUrl": "", *//Path to the repo hosting the scripts in Stash. (e.g. CDL/mimir-scripts). Leave empty to use the default*

"requisiteStageRefIds": [

"10"

],

"scriptPath": "", *//Path to the folder hosting the scripts in Stash. (e.g. groovy, python or shell)*

"type": "script",

"user": "[anonymous]",

"waitForCompletion": true *//if unchecked, marks the stage as successful right away without waiting for the script to complete*

},

## 9-> WebHook *// Webhook Runs a Webhook job*

{

"method": "POST",

"name": "Webhook",

"refId": "12",

"requisiteStageRefIds": [

"11"

],

```
  "statusUrlResolution": "getMethod",
  "type": "webhook",
  "url": ""
}
```