

Infrastructure Unknown

Using Polymorphic Templates To Predict The Future

Sean Davis

@seanasaservice

Sr. Business Transformation Consultant, DevOps Lead

@AgileThought



Been In IT For 20+ Years
Love Pretty Much Anything That Goes Fast
Have a Heart For Helping Others
Father To Several Wayward Children

Community Involvement

- DevOps Institute / ITSM - DevOps / DevSecOps Instructor
- Nationwide Public Speaker
- Pluralsight Author
- Executive / Life Coach
- Provide Leadership Courses For Gifted & Young Adults



Agenda

- Static Templates
- Polymorphic Templates
- Polymorphic Methods
- Method Examples
- Wrap Up / Resources

Static Templates



- All Application Templates And Settings Are Static One Off's
- Cloned Applications / Other Environments Could Have Drift From Original
- No Template Governance From Security Perspective
- Most Values Will Be Hardcoded For the Specific App Or Environment
- Massive Template Sprawl Will Occur Over Time For Self Service Catalogs
- Lessons Learned From Improvements Are Lost In Individual Application Code Repos
- Templates Sometimes Require Heavy Customization To Be Reused

Polymorphic Templates

Reference Resource Templates From
Our Master Template

Deploy Infrastructure and App Code

Embed Dynamic Variables & Overrides
Into Master Template or Code

The Capacity To Take On Different Forms

Our Discussion: The term polymorphism is used to

- Application Templates Indicate how Changes
- Cloned Applications Will Have Same Config As Original
- Templates Can Be Governed Across Entire Business
- Most Values Will Be Dynamically Generated
- Minimal Template Sprawl Due To Common Components Templates
- All Applications Gain From Improvements In Individual Templates
- Microservice Templates Can Be Reused By Anyone With Little Effort

App A Master Template

WebApp
Template
Overrides

DBaaS
Template
Overrides

WebApp Template

Security
Standards

Default
Configs

Template
Boundaries
(Locks,
Limits, Etc.)

Dynamic
Values

DBaaS Template

Security
Standards

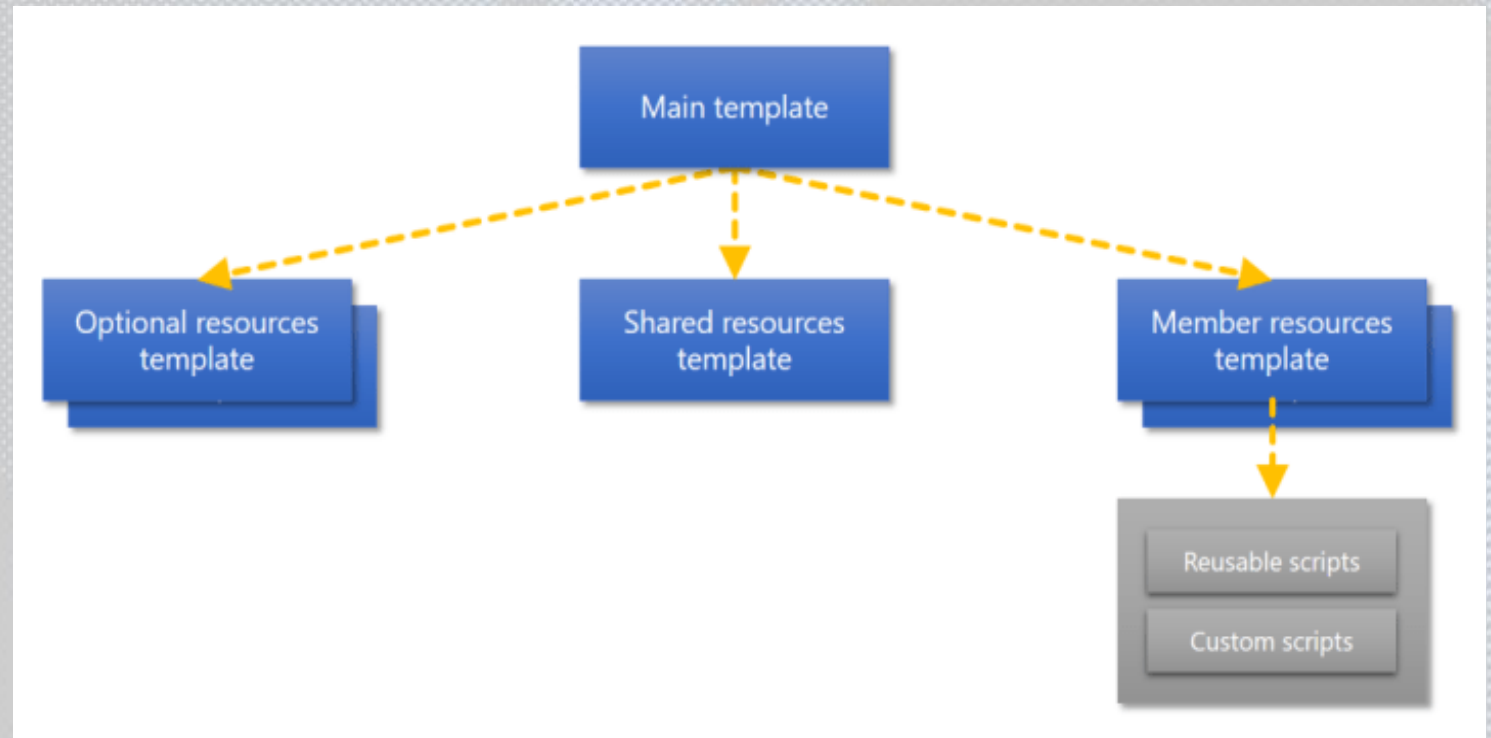
Default
Configs

Template
Boundaries
(Locks,
Limits, Etc.)

Dynamic
Values

Polymorphic Methods

- Methods
 - Nested / Linked Templates
 - Dynamic / Calculated Variables
 - Template Functions
 - Scoped Parameters
- Practices (Assist Poly Templates)
 - Shared Resource Templates
 - Reusable Scripts



Scoped Parameters Example

```
"parameters": {  
  "storageSKU": {  
    "type": "string",  
    "allowedValues": [  
      "Standard_LRS",  
      "Standard_ZRS",  
      "Standard_GRS",  
      "Standard_RAGRS",  
      "Premium_LRS"  
    ],  
    "defaultValue": "Standard_LRS",  
    "metadata": {  
      "description": "The type of replication to use for the storage account."  
    }  
  }  
}  
"resources": [  
  {  
    "type": "Microsoft.Storage/storageAccounts",  
    "sku": {  
      "name": "[parameters('storageSKU')]"  
    }  
  }  
]
```

Dynamic / Calculated Variables Example

```
"parameters": {
  "environmentName": {
    "type": "string",
    "allowedValues": [
      "test",
      "prod"
    ]
  }
},
"variables": {
  "storageName": "[concat(toLower(parameters('environmentName')), uniqueString(resourceGroup().id))]"
},
"resources": [
  {
    "apiVersion": "2015-08-01",
    "type": "config",
    "name": "connectionstrings",
    "properties": {
      "AzureWebJobsDashboard": {
        "value":
          "[concat('DefaultEndpointsProtocol=https;AccountName=',variables('storageName'),';;AccountKey=',concat(listKeys(variables('storageName'),'2015-05-01-preview').key1))]",
        "type": "Custom"
      }
    }
  }
]
```

Calculated Value

Dynamic Value

Template Function Example

```
"functions": [
  {
    "namespace": "contoso",
    "members": {
      "uniqueName": {
        "parameters": [
          {
            "name": "storageNamePrefix",
            "type": "string"
          }
        ],
        "output": {
          "type": "string",
          "value": "[concat(toLower(parameters('storageNamePrefix')), uniqueString(resourceGroup().id))]"
        }
      }
    }
  }
],
"resources": [
  {
    "name": "[contoso.uniqueName(parameters('storageNamePrefix'))]",
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2016-01-01",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "location": "South Central US",
    "tags": {},
    "properties": {}
  }
]
```


Nested / Linked Template Example

```
"resources": [  
  {  
    "apiVersion": "2017-05-10",  
    "name": "linkedTemplate",  
    "type": "Microsoft.Resources/deployments",  
    "properties": {  
      "mode": "Incremental",  
      "templateLink": {  
        "uri": "https://mystorageaccount.blob.core.windows.net/AzureTemplates/newStorageAccount.json",  
        "contentVersion": "1.0.0.0"  
      },  
      "parametersLink": {  
        "uri": "https://github.com/imseandavis/AzureTemplates/newStorageAccount.parameters.json",  
        "contentVersion": "1.0.0.0"  
      }  
    }  
  }  
]
```


Wrap Up / Resources

- Provides Template Reusability Across The Business
- Provides Greater Security Control And Practices For Application Templates
- Breaks Complex Templates Down Into Modular And Commoditized Portions
- Reduces Rework Due To Troubleshooting Template Changes Or Out Of Band Values
- Check Out The QR Code / Link Below For Links And Resources On Today's Examples



<https://bit.ly/2Admu9Z>

Thank You All Day DevOps Sponsors

Platinum Sponsors



Gold Sponsors



Media Sponsors

Thank You All Day DevOps Supporters





Meet me in the Slack channel for Q&A

bit.ly/addo-slack