Orchestrating the Cloud

Matt Wood

TECHNOLOGY EVANGELIST

Welcome

AGENDA

Orchestrating the Cloud

- ☐ 1. Application architecture
- ☐ 2. Role of orchestration
- ☐ 3. Pillars of orchestration
- ☐ 4. Orchestration by example
- ☐ 5. Summary

1

Application Architecture

Applications in the cloud

3 tiers

Code

Configuration

Application tier

Code

Configuration

Application tier

Application tier

Code

Configuration

Service tier

Operating system

Integration settings

Launch configuration

Services + configuration

Application tier

Code

Configuration

Service tier

Operating system

Integration settings

Launch configuration

Services + configuration

Application tier

Code

Configuration

Service tier

Operating system

Integration settings

Launch configuration

Services + configuration

Infrastructure tier

AMIs

Architecture

Multi-AZ

Scaling rules

Security groups

Middleware

Value baked into each tier

Value in application

Value in service tier

Optimisation

Configuration

Value in service tier

Technology choices

Value in infrastructure

Engine room

Optimised

Value in infrastructure

Scalable

Fault tolerant

Orchestration maximises this value

Ephemeral

to

concrete

One team to whole organisation

One hit to reproducible

Brittle to strong

Maximise value

Minimise risk

2

Role of Orchestration

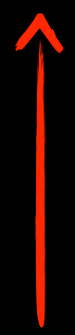
Cloud life cycle

Initialisation

Steady state run time

Updates

Application updates



Updates

Service updates

Scale events

Change management

Very meta!

Managing change management

3

Pillars of Orchestration

ZEROTH PILLAR

Version control

F I R S T P I L L A R

Provisioning orchestration

CloudFormation

aws.amazon.com/cloudformation

Template

Define a full infrastructure stack

Auto-scaling

RDS

EC2

SNS

SimpleDB

SQS

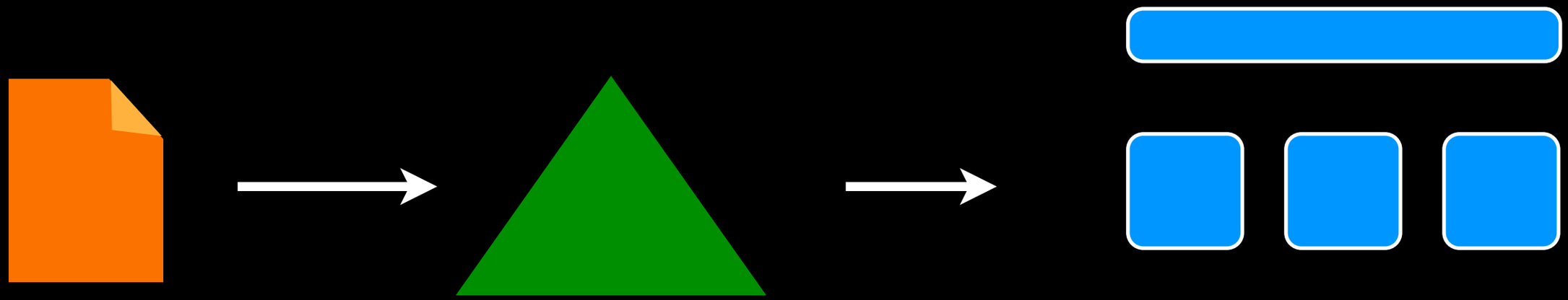
Resources

Elastic Beanstalk

CloudWatch

Security groups

Tags



Template

CloudFormation

Provisioned resources

Complete definition

Atomic

Idempotent

Free

Anatomy of a template

JSON

Plain text

Perfect for version control

JSON

Validate-able

Declarative language

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "Create an EC2 instances",

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instance",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
          "AMI" : "ami-76f0061f"
      },
      "us-west-1" : {
          "AMI" : "ami-655a0a20"
      },
      "eu-west-1" : {
          "AMI" : "ami-7fd4e10b"
      },
      "ap-southeast-1" : {
          "AMI" : "ami-72621c20"
      },
      "ap-northeast-1" : {
          "AMI" : "ami-8e08a38f"
      }
    }
  },

  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  },

  "Outputs" : {
    "InstanceId" : {
      "Description" : "InstanceId of the newly created EC2 instance",
      "Value" : { "Ref" : "Ec2Instance" }
    },
    "AZ" : {
      "Description" : "Availability Zone of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone" ] }
    },
    "PublicIP" : {
      "Description" : "Public IP address of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicIp" ] }
    }
  }
}
```

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "Create an EC2 instances",

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instance",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
          "AMI" : "ami-76f0061f"
      },
      "us-west-1" : {
          "AMI" : "ami-655a0a20"
      },
      "eu-west-1" : {
          "AMI" : "ami-7fd4e10b"
      },
      "ap-southeast-1" : {
          "AMI" : "ami-72621c20"
      },
      "ap-northeast-1" : {
          "AMI" : "ami-8e08a38f"
      }
    }
  },

  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  },

  "Outputs" : {
    "InstanceId" : {
      "Description" : "InstanceId of the newly created EC2 instance",
      "Value" : { "Ref" : "Ec2Instance" }
    },
    "AZ" : {
      "Description" : "Availability Zone of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone" ] }
    },
    "PublicIP" : {
      "Description" : "Public IP address of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicIp" ] }
    }
  }
}
```

Headers
Parameters

Mappings

Resources

Outputs

Parameters

Provision-time specification
Command line options

```
"Parameters" : {
  "KeyName" : {
    "Description" : "Name of an existing
     EC2 KeyPair to enable SSH access to
     the instance",
    "Type" : "String"
  }
},
```

Mappings

Conditionals
Case statements

```
"Mappings" : {
  "RegionMap" : {
    "us-east-1" : {
        "AMI" : "ami-76f0061f"
    },
    "us-west-1" : {
        "AMI" : "ami-655a0a20"
    },
    "eu-west-1" : {
        "AMI" : "ami-7fd4e10b"
    },
    "ap-southeast-1" : {
        "AMI" : "ami-72621c20"
    },
    "ap-northeast-1" : {
        "AMI" : "ami-8e08a38f"
    }
  }
},
```

```
"Mappings": {
  "AWSInstanceType2Arch" : {
    "t1.micro"    : { "Arch" : "64" },
    "m1.large"    : { "Arch" : "64" },
    "m1.xlarge"   : { "Arch" : "64" },
    "m2.xlarge"   : { "Arch" : "64" },
    "m2.2xlarge"  : { "Arch" : "64" },
    "m2.4xlarge"  : { "Arch" : "64" },
    "c1.xlarge"   : { "Arch" : "64" },
    "cc1.4xlarge" : { "Arch" : "64" }
  },
```

Resources

```
"Resources" : {
  "Ec2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : { "Fn::FindInMap" :
[ "RegionMap", { "Ref" : "AWS::Region" },
"AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  }
```

```
"Resources" : {
  "Ec2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : { "Fn::FindInMap" :
[ "RegionMap", { "Ref" : "AWS::Region" },
"AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  }
```

```
"Resources" : {
  "Ec2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : { "Fn::FindInMap" :
[ "RegionMap", { "Ref" : "AWS::Region" },
"AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  }
```

```
"KeyName" : { "Ref" : "KeyName" },
```

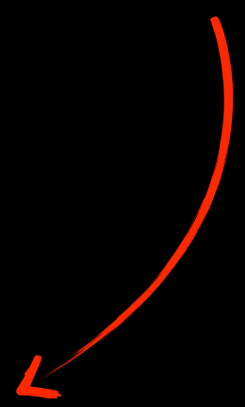
Parameter reference

```
"ImageId" : {

  "Fn::FindInMap" :
  [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]

},
```

Map conditional



```
"ImageId" : {

   "Fn::FindInMap" :
   [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]

},
```

```
"ImageId" : {
  "Fn::FindInMap" :
    [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]
},
```

Name of map

```
"ImageId" : {
  "Fn::FindInMap" :
    [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]
},
```

Intrinsic property reference

Outputs

Returned values

```
"Outputs" : {
  "InstanceId" : {
    "Description" : "InstanceId of the newly created EC2 instance",
    "Value" : { "Ref" : "Ec2Instance" }
  },
  "AZ" : {
    "Description" : "Availability Zone of the newly created EC2 instance",
    "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone" ] }
  },
  "PublicIP" : {
    "Description" : "Public IP address of the newly created EC2 instance",
    "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicIp" ] }
  }
 }
}
```

Deliver via API

Validate via API

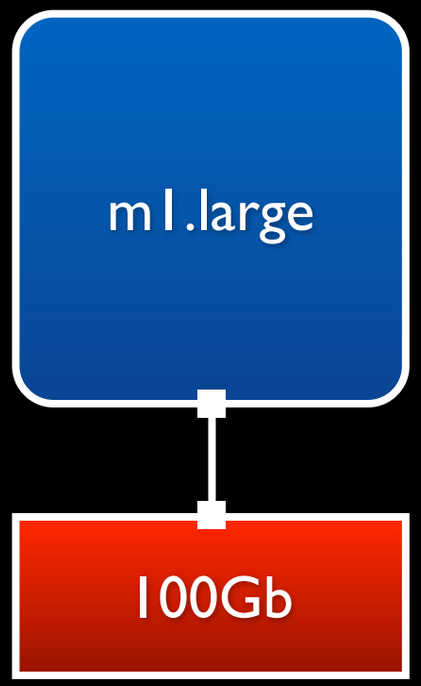
Deliver via S3

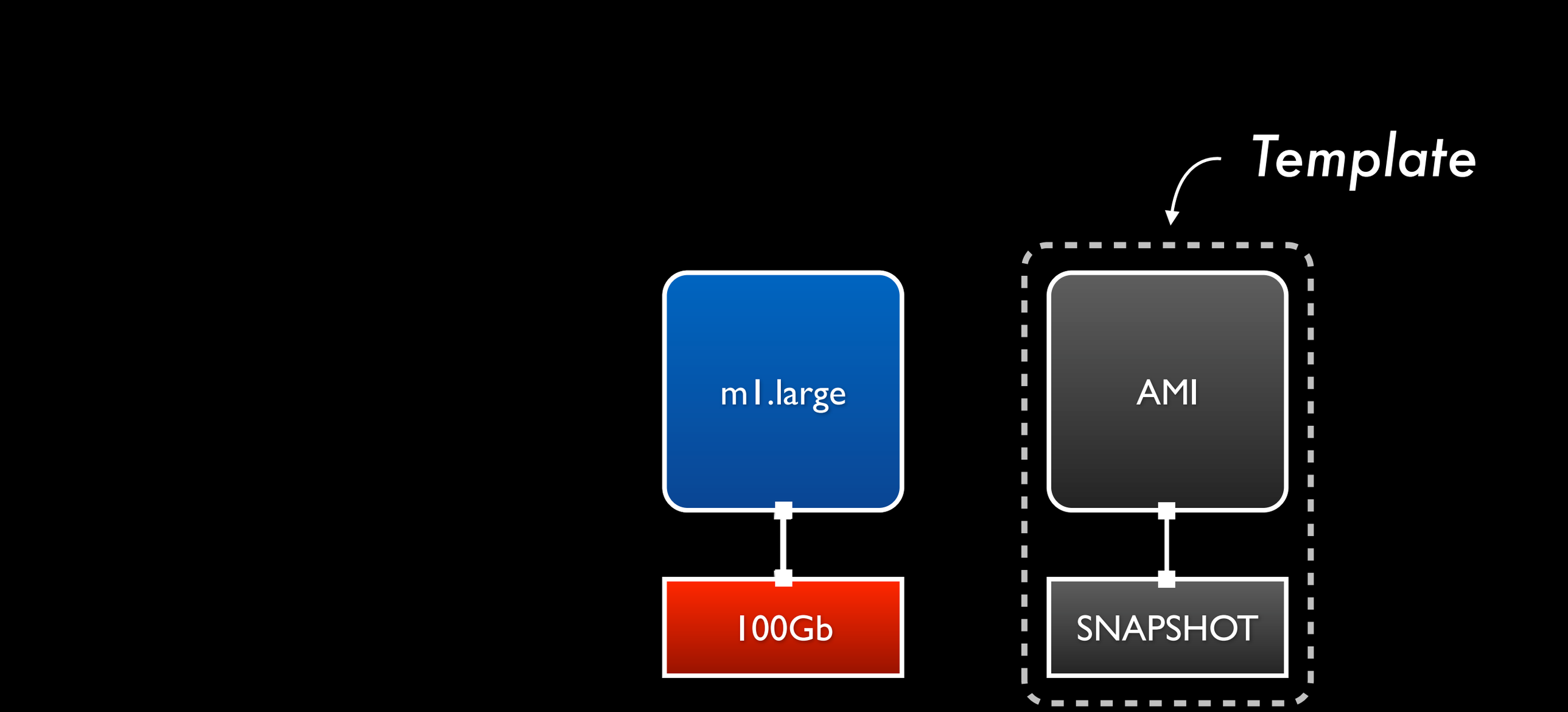
Growing library

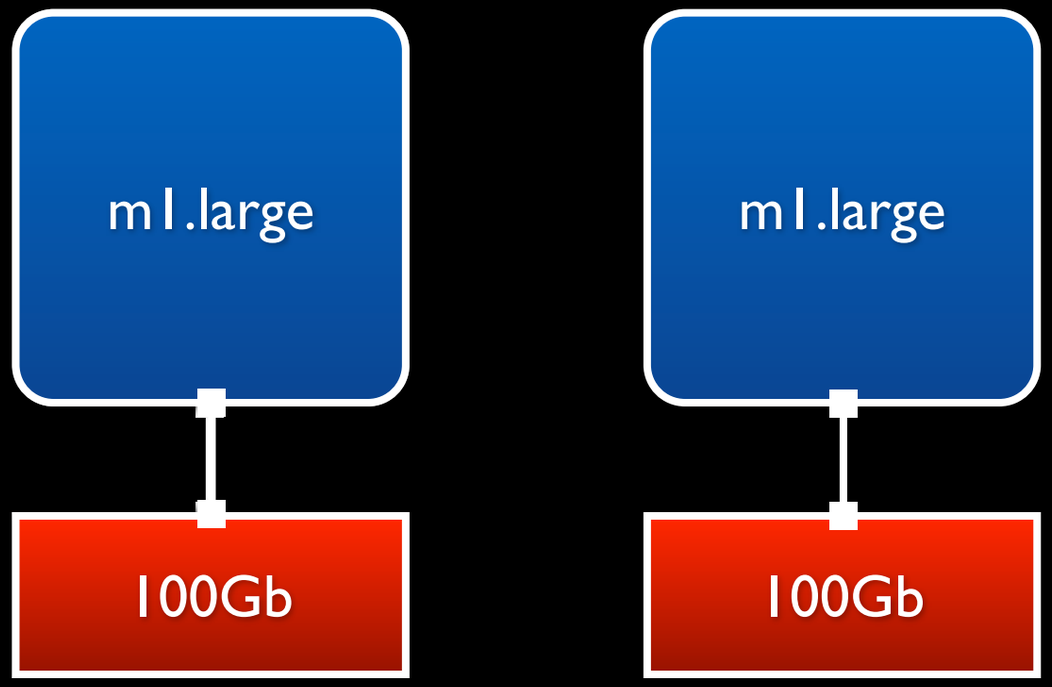
SECOND PILLAR

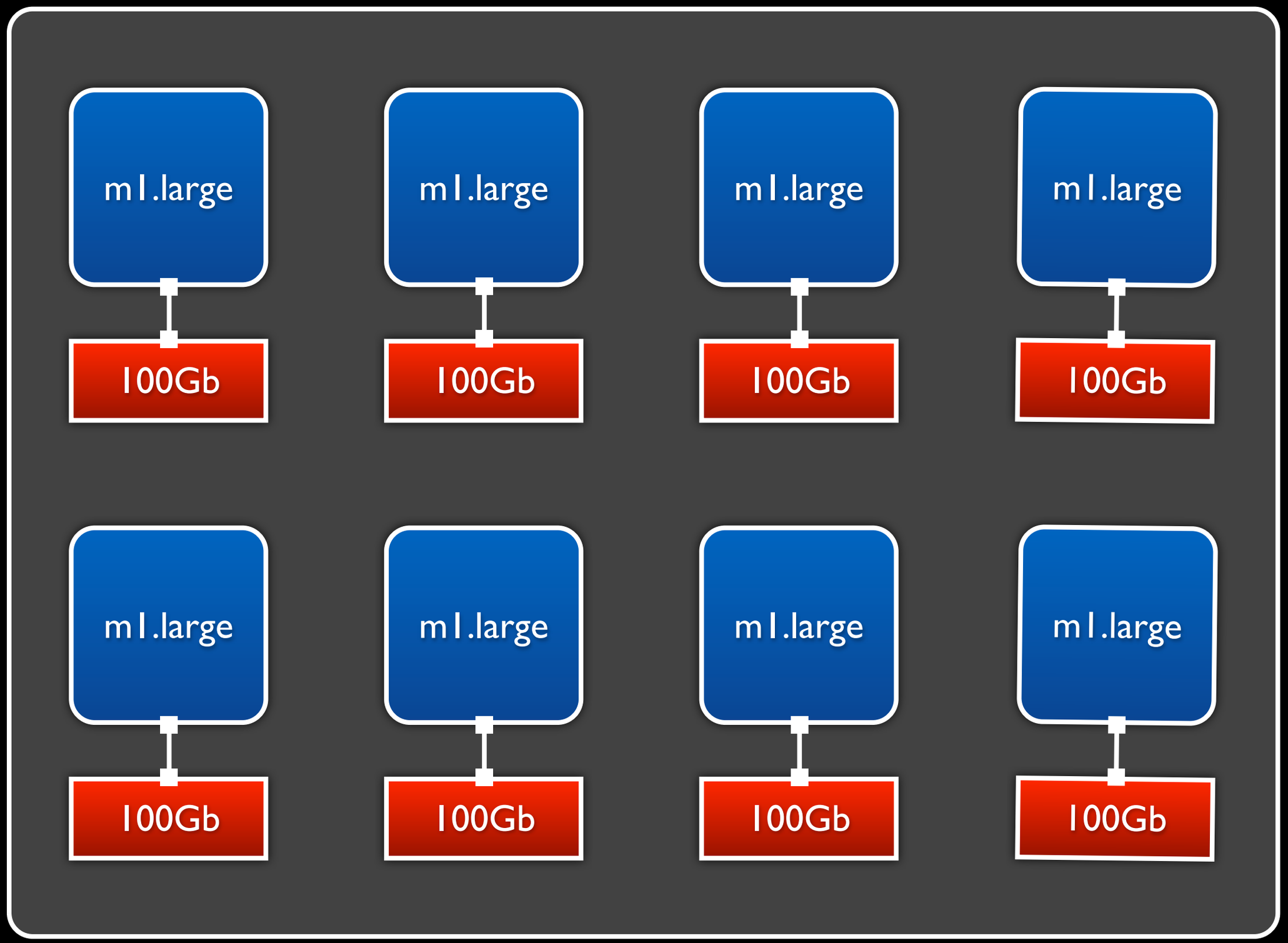
Configuration management

Custom AMI


m1.large
100Gb


Template
m1.large
AMI
100Gb
SNAPSHOT


m1.large
m1.large
100Gb
100Gb


m1.large
m1.large
m1.large
m1.large
100Gb
100Gb
100Gb
100Gb
m1.large
m1.large
m1.large
m1.large
100Gb
100Gb
100Gb
100Gb

Bootstrap

Generic AMI

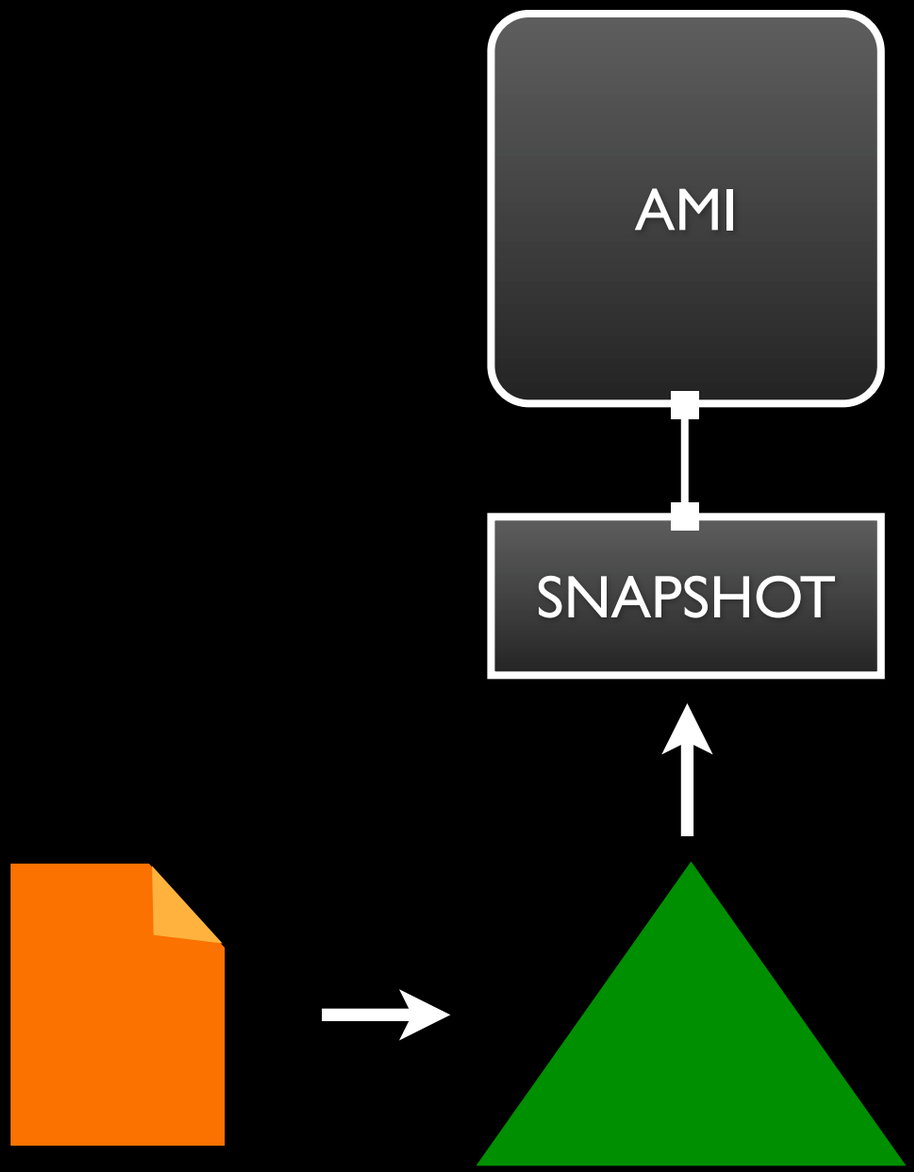
Custom build

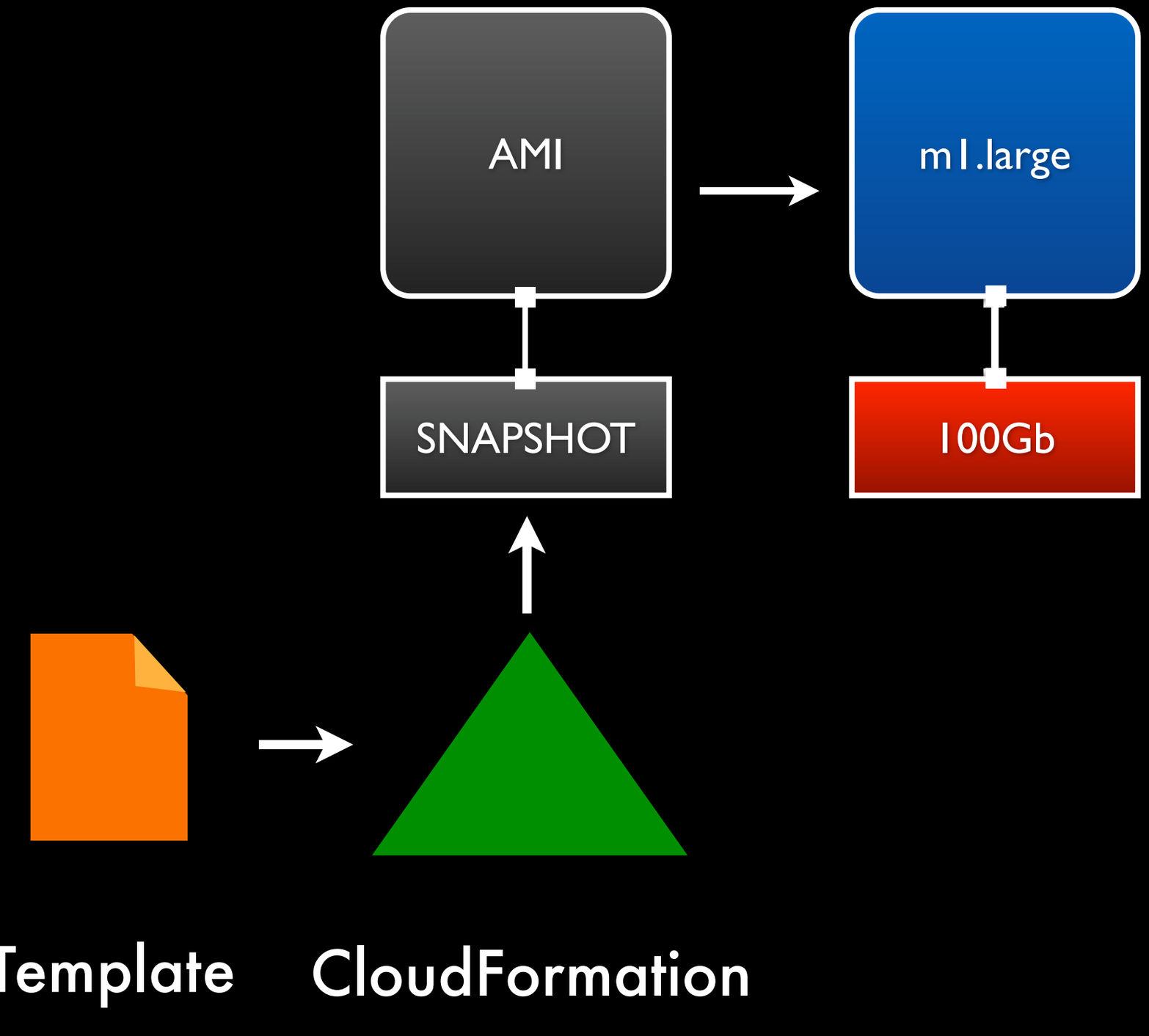
Services *Dependencies*

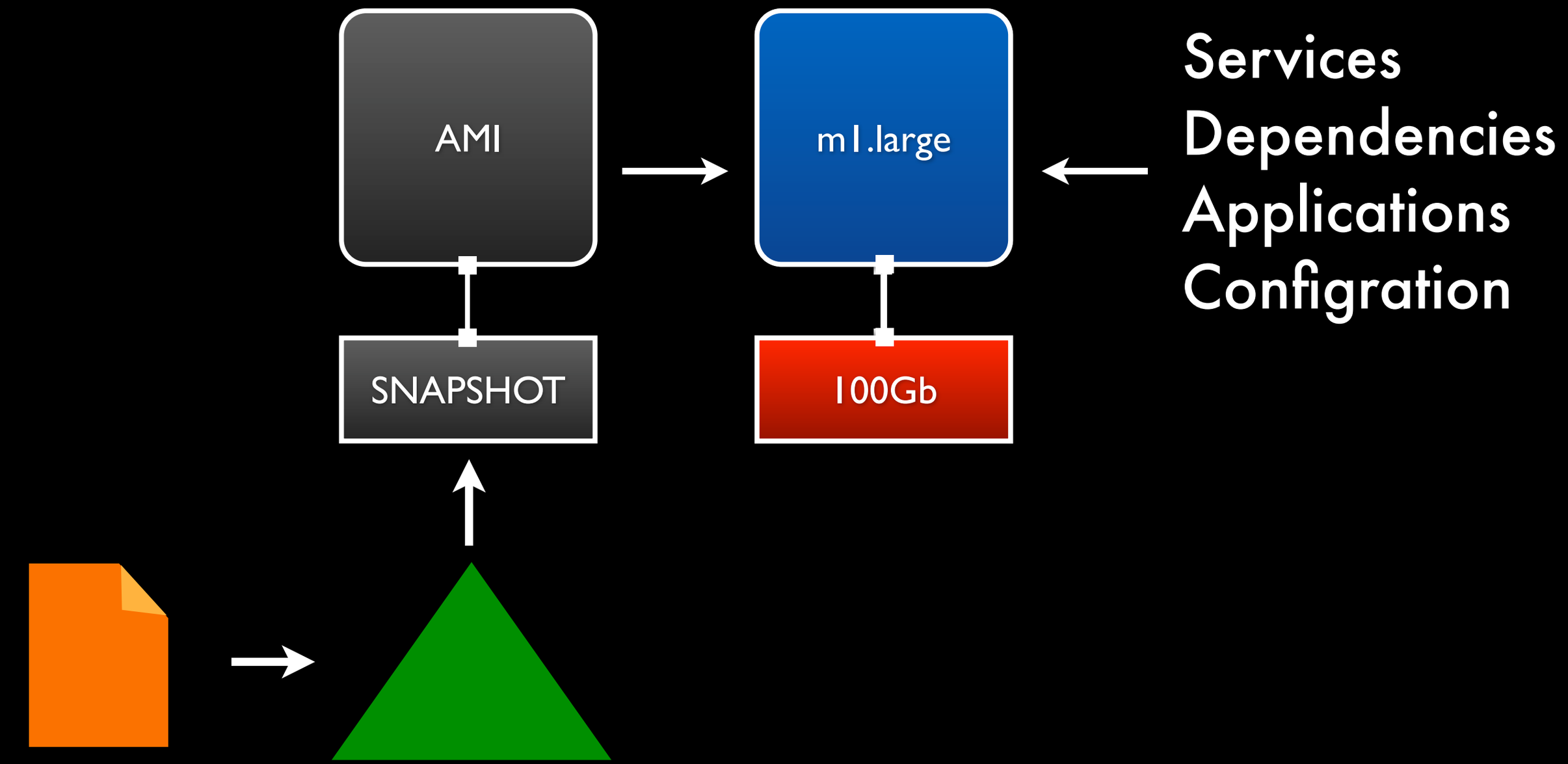
Define manifests

Configuration

Applications


AMI
SNAPSHOT
Template
CloudFormation

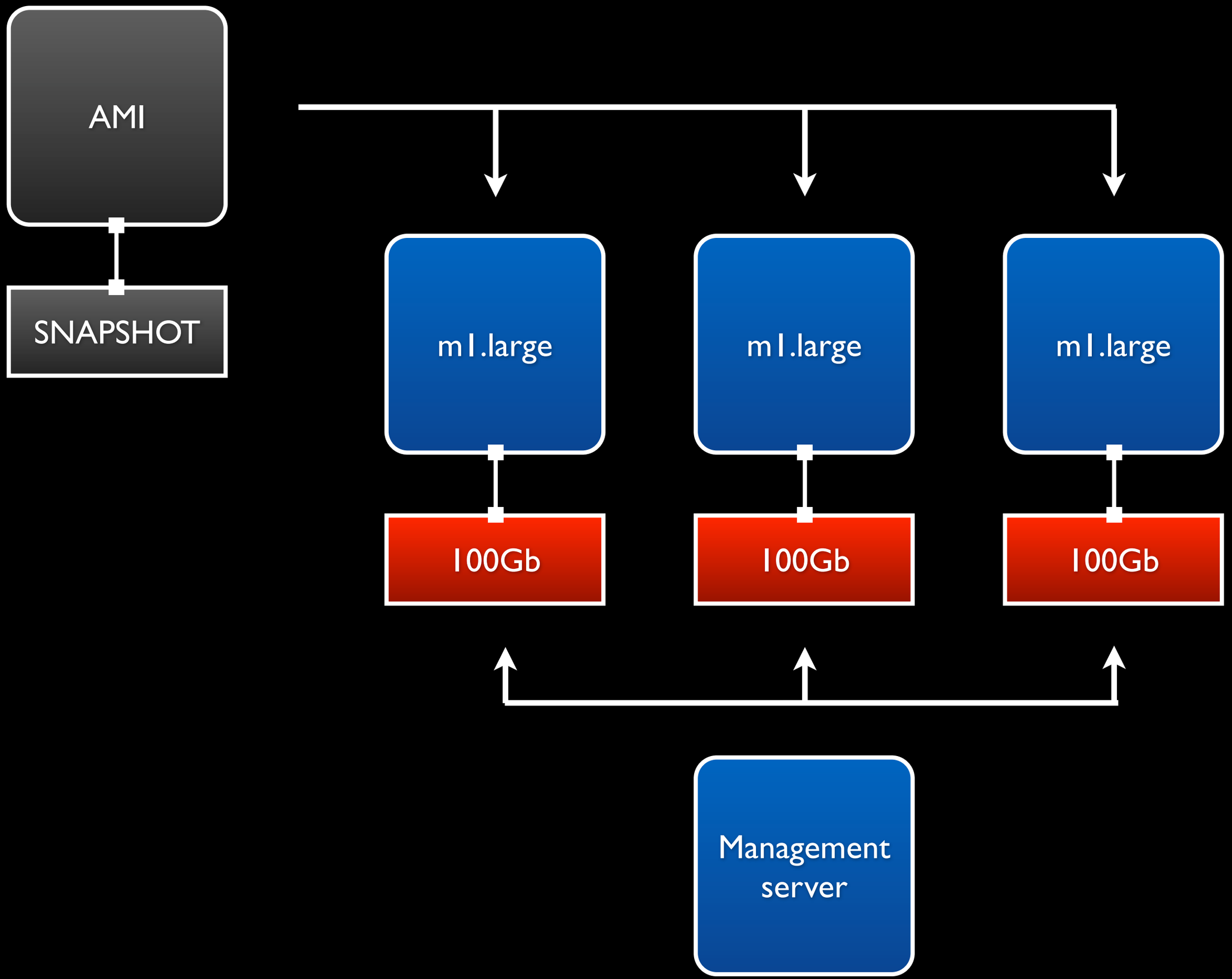

AMI
m1.large
SNAPSHOT
100Gb
Template
CloudFormation


AMI
m1.large
Services
Dependencies
Applications
Configration
SNAPSHOT
100Gb
Template
CloudFormation

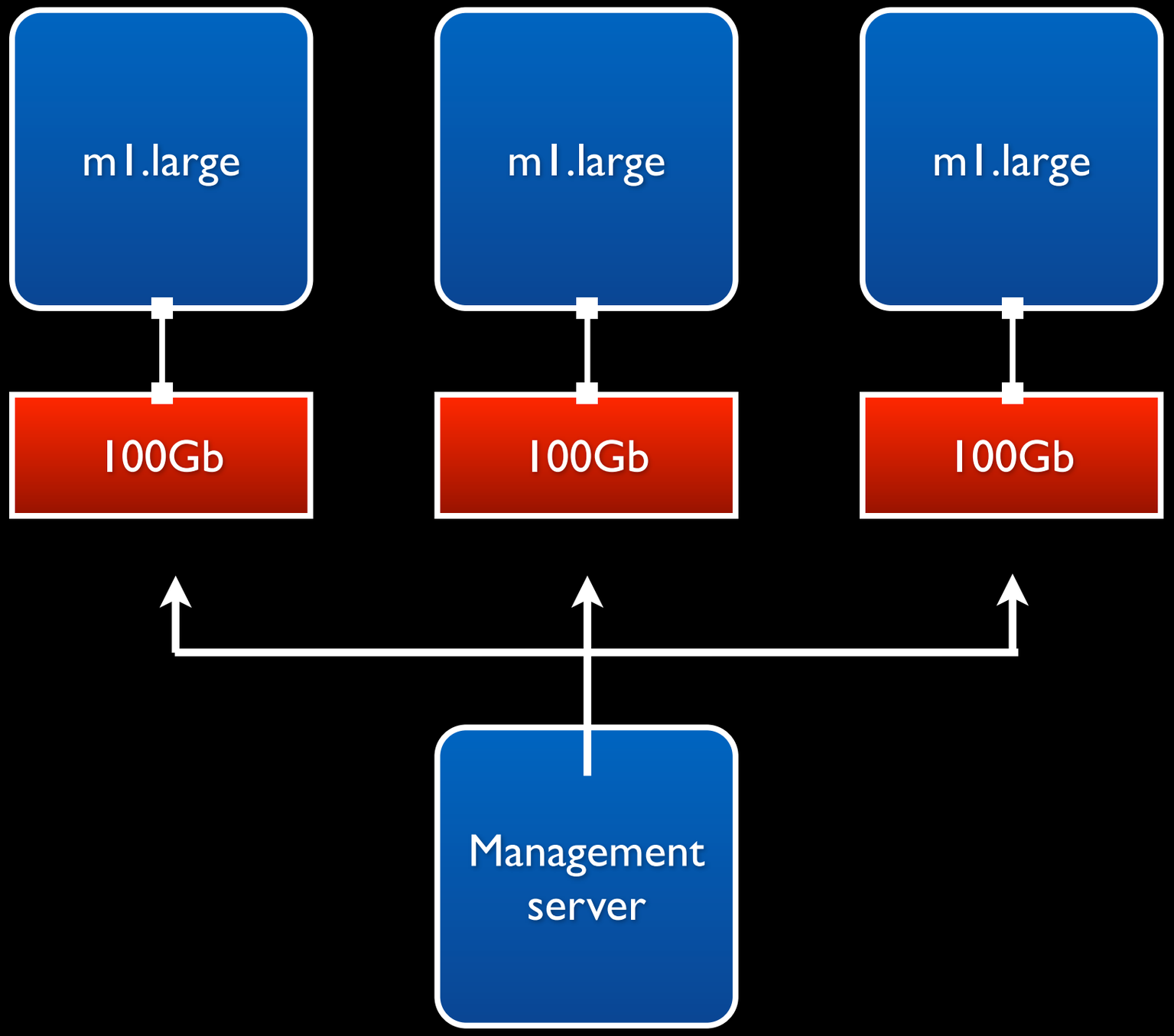
1. Setup users and groups
2. Install Apache
3. Configure Apache
4. Setup directories
5. Start ancillary services
6. Deploy code

Management server

Pull


AMI
SNAPSHOT
m1.large
m1.large
m1.large
100Gb
100Gb
100Gb
Management server

Push


m1.large
m1.large
m1.large
100Gb
100Gb
100Gb
Management server

Fewer AMIs to manage

Versioned configuration

Codified updates

Known state

Rolling updates

Simulations

Built for elastic architectures

Loose coupling

Address via meta-data

And much more!

:(

Extra overhead

Chef

+ Knife

Puppet

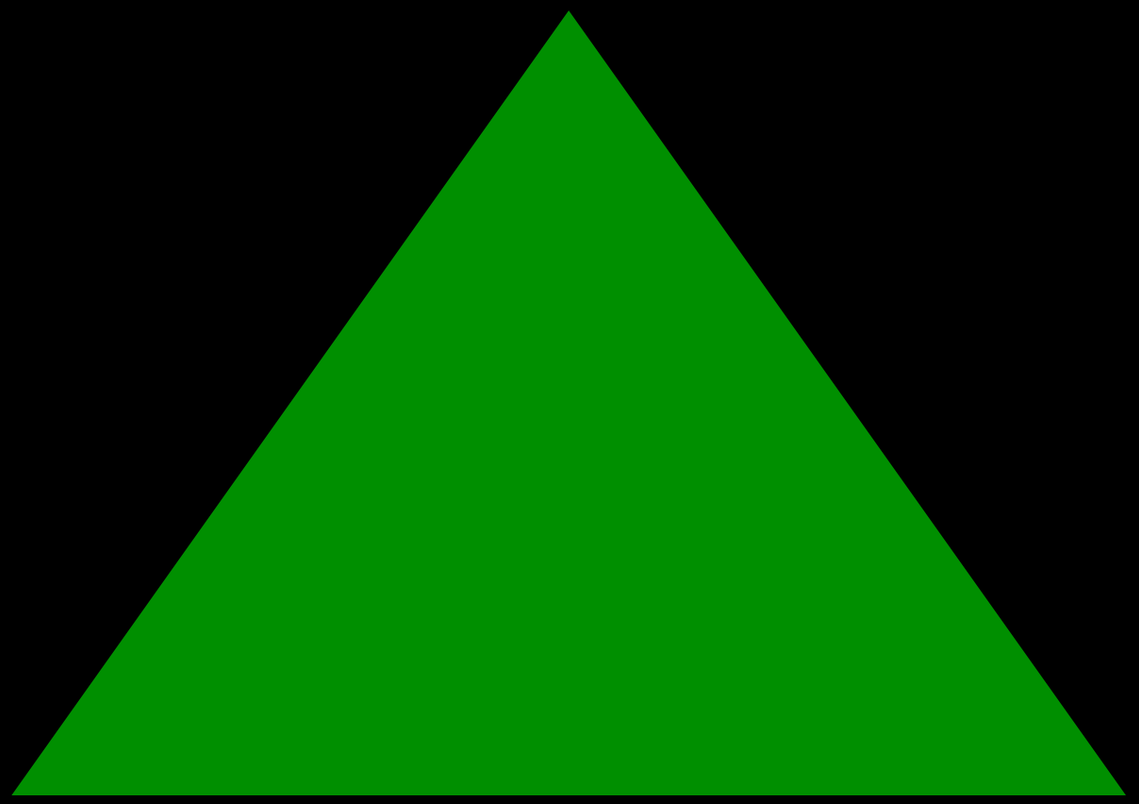
+ MCollective

THIRD PILLAR

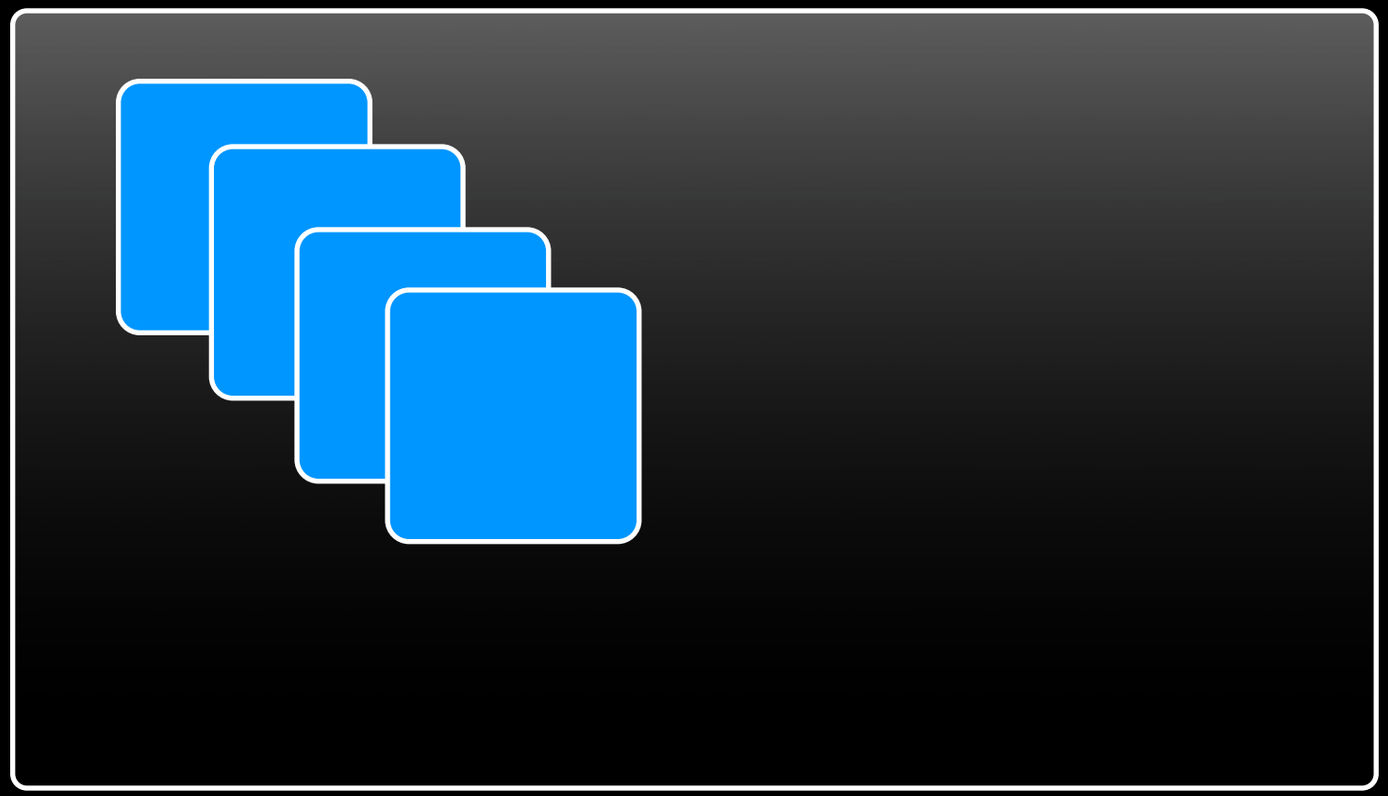
Performance automation

Auto-scaling

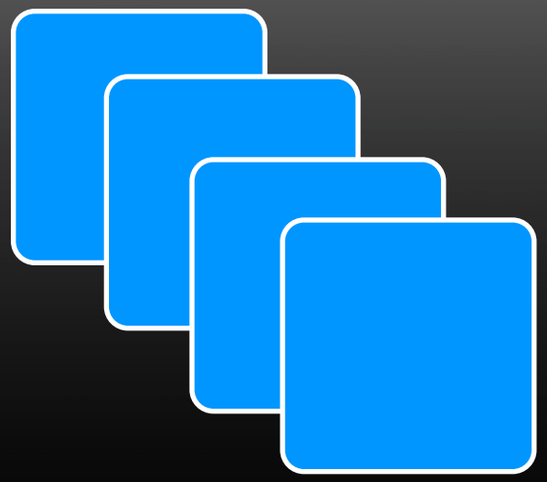
ELB



CloudWatch Auto-scaling



Scaling group



Scaling group

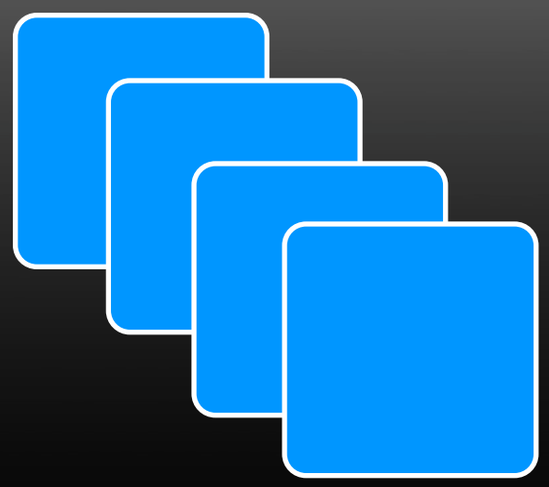
DatabaseConnections

DatabaseConnections

Triggers

(Alarms + Policies)

Launch
configurations



Scaling group

`DatabaseConnections`

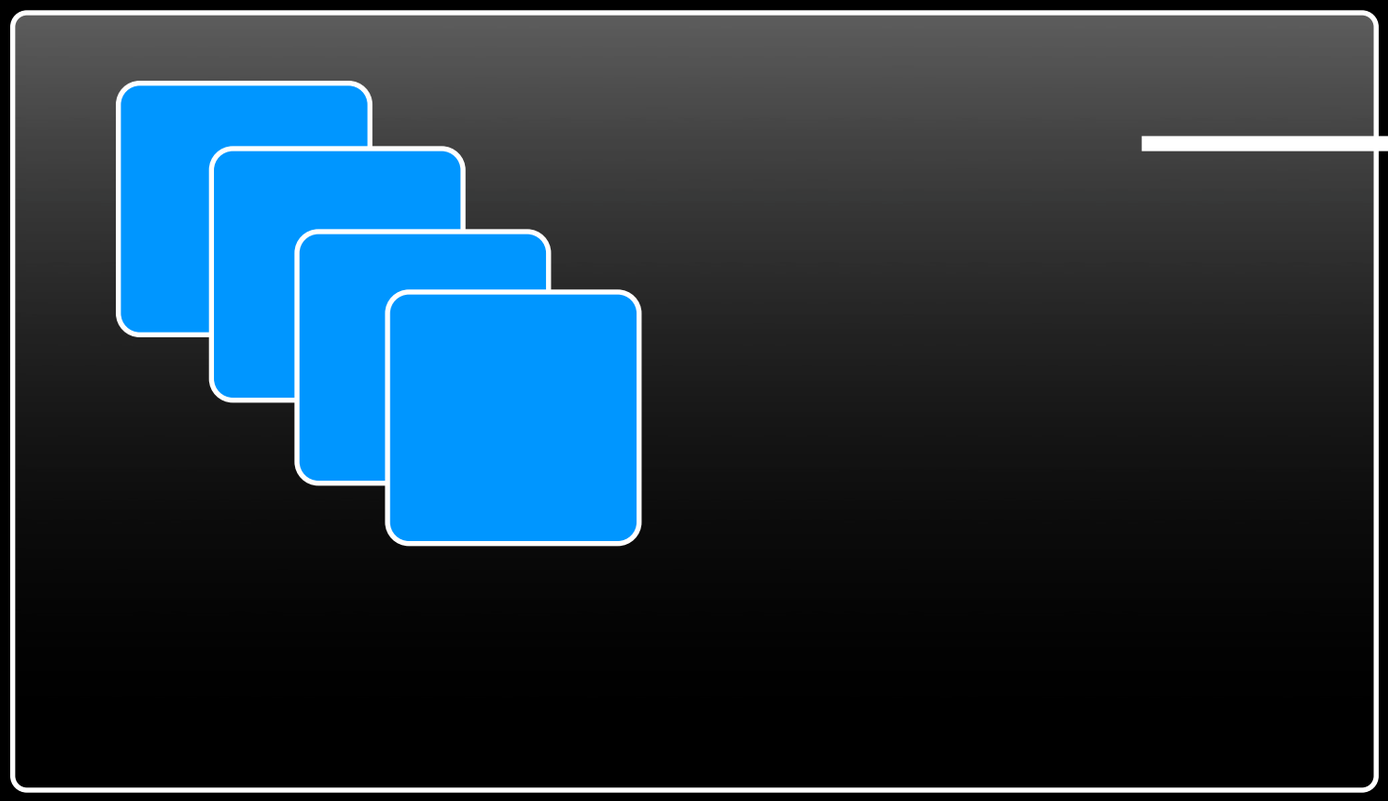
`DatabaseConnections`

Triggers

(Alarms + Policies)

Additional performance

Launch
configurations

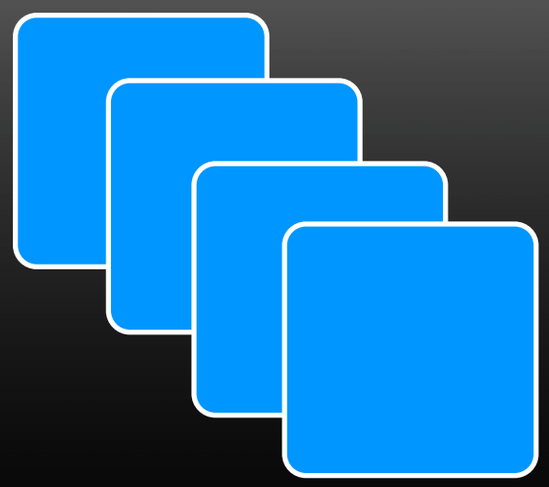


Scaling group

Triggers

(Alarms + Policies)

Launch
configurations



Scaling group

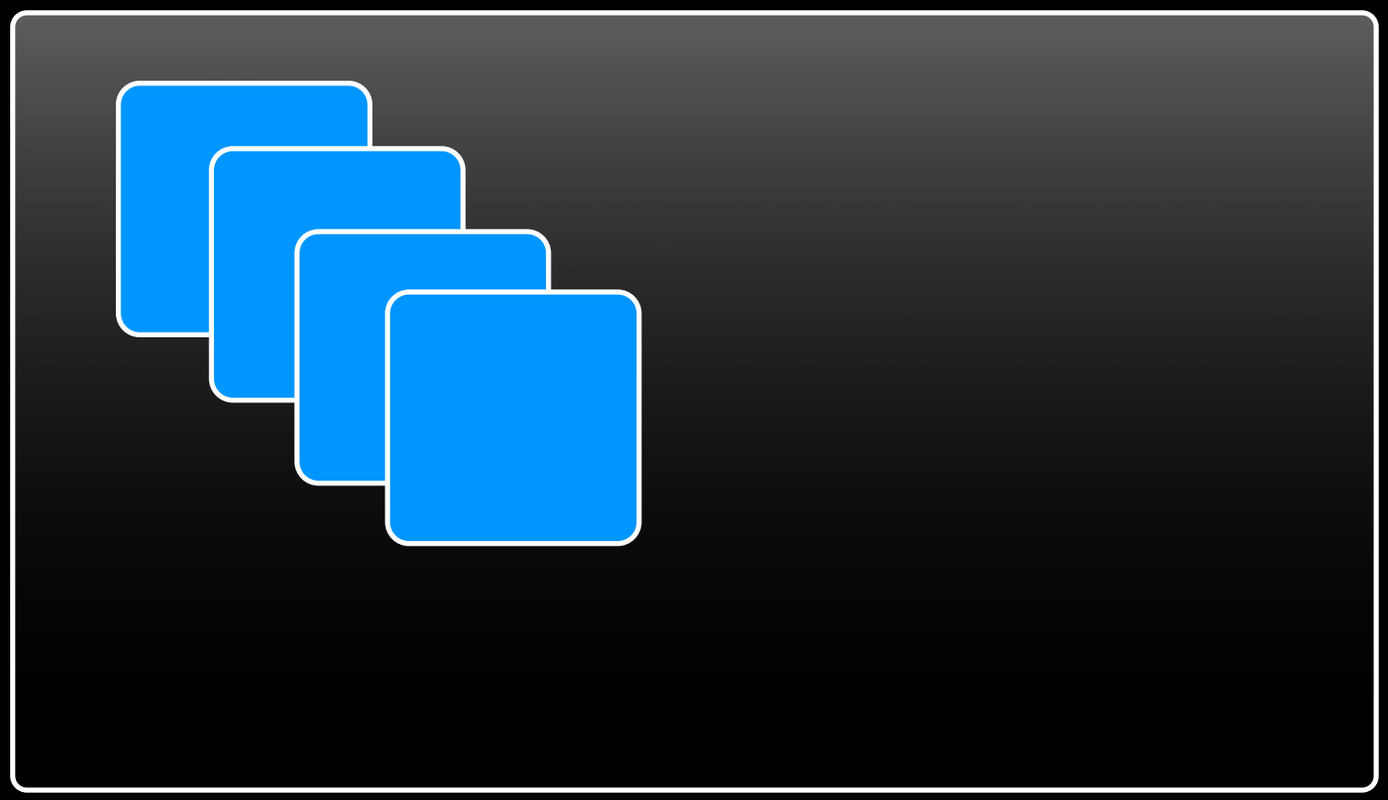
`DatabaseConnections`

`DatabaseConnections`

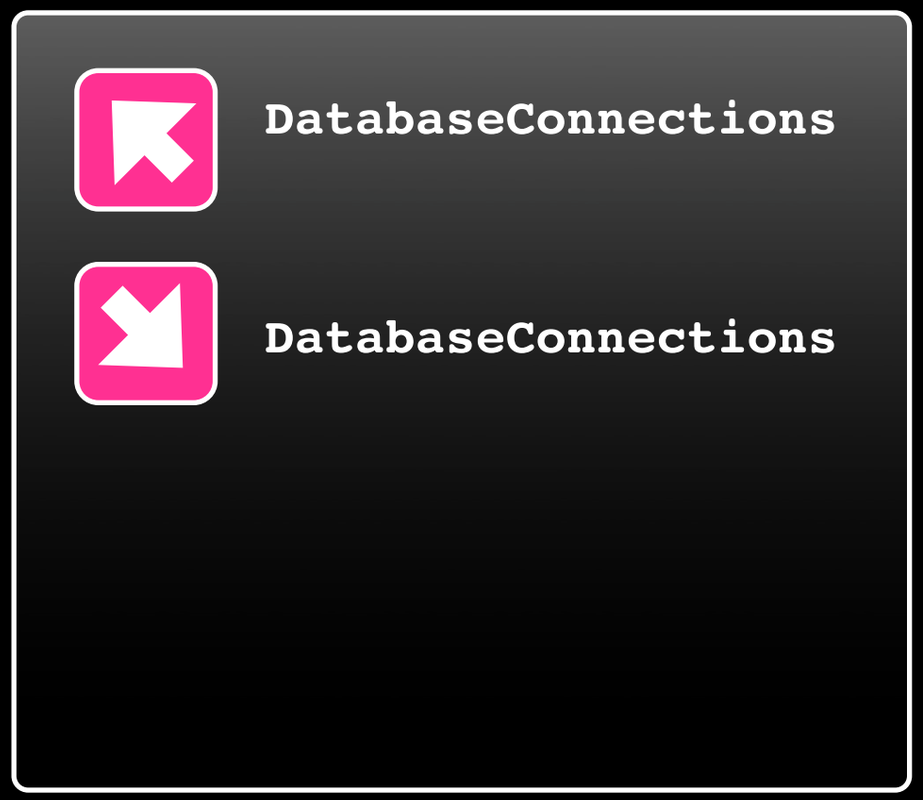
Triggers

(Alarms + Policies)

Launch
configurations



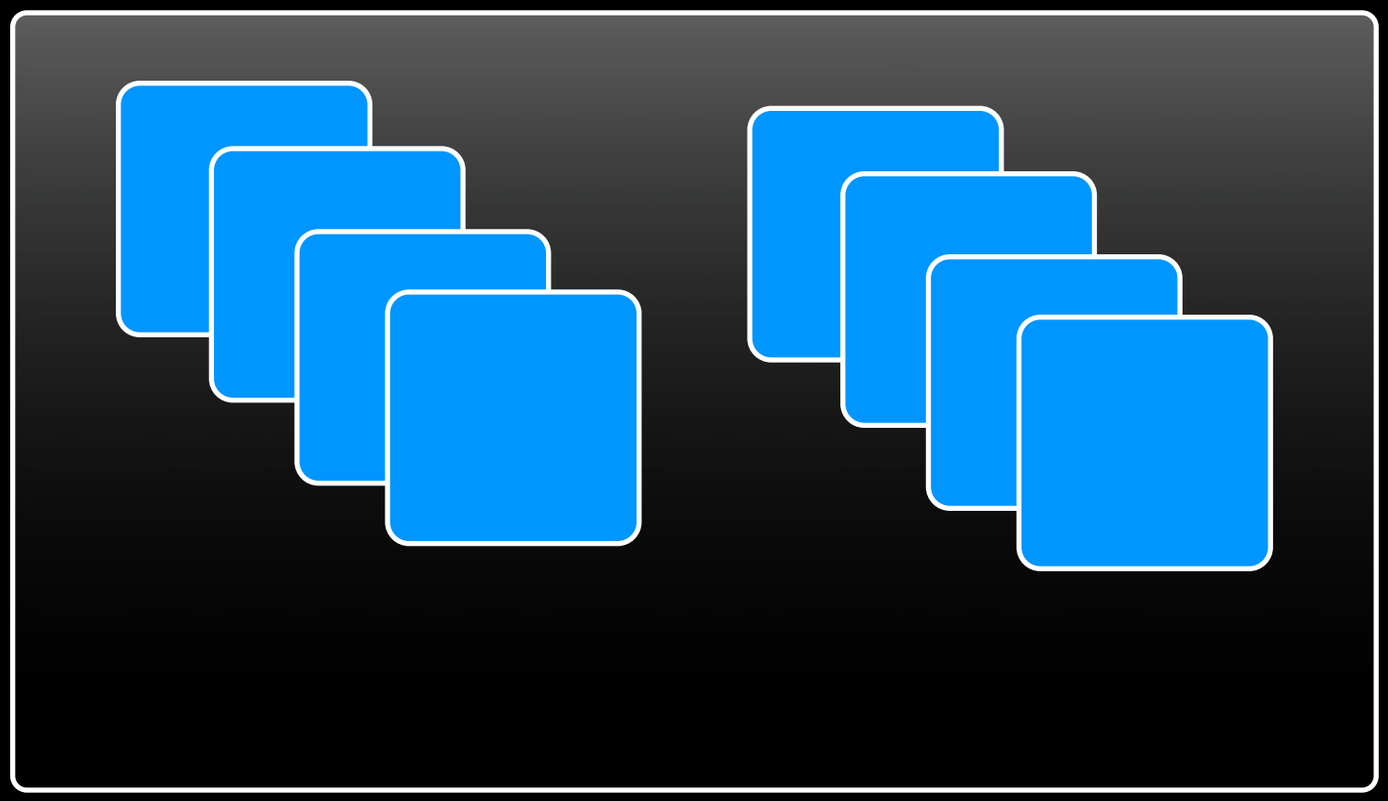
Scaling group



Triggers

(Alarms + Policies)

Launch
configurations



Scaling group

Triggers

(Alarms + Policies)

Auto-healing

4

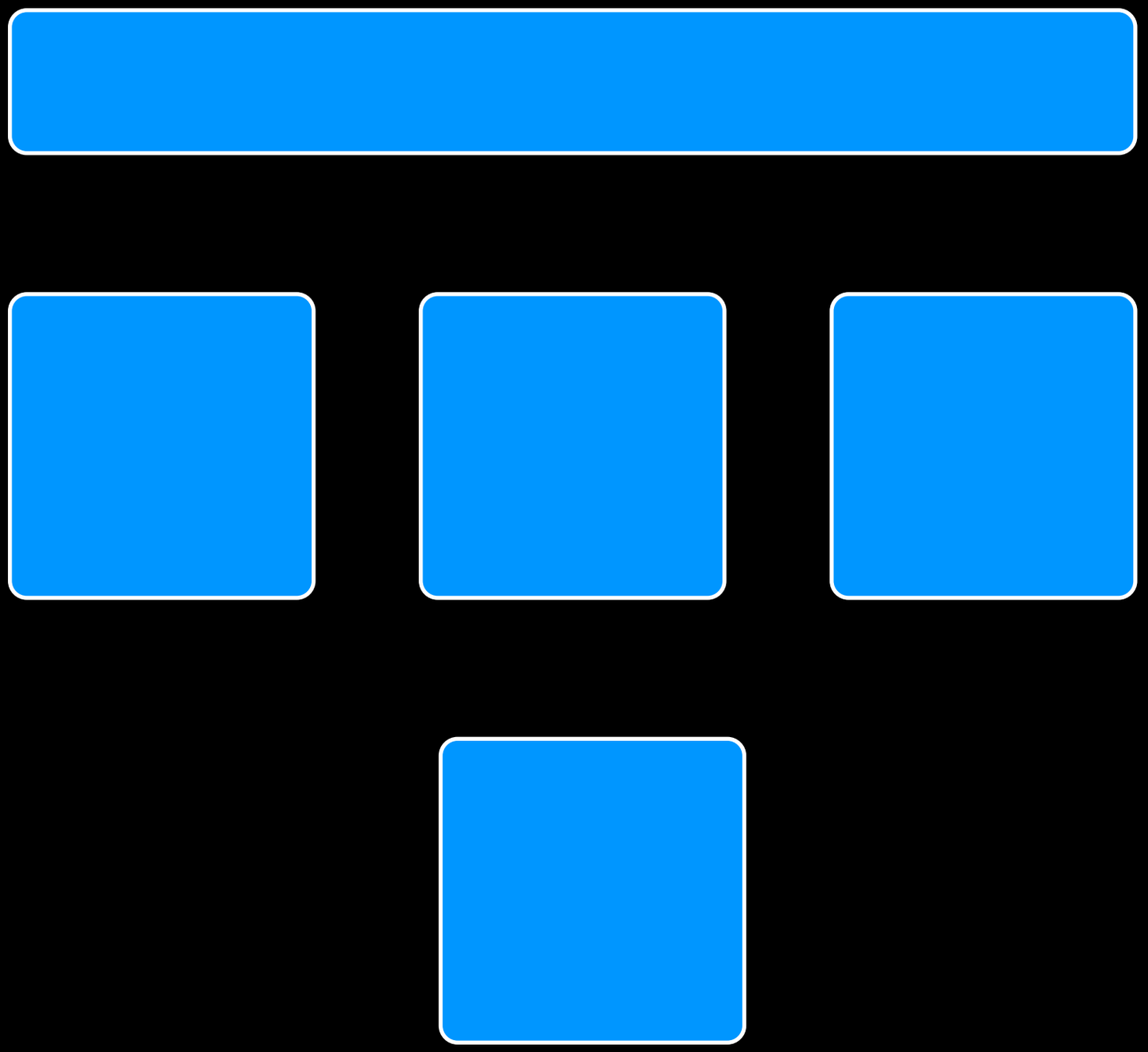
Orchestration by Example

Web application

Initialisation

with CloudFormation

Design stack



Load balancer

Fault tolerant web servers

RDS

Create template

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",


  "Parameters" : {
    "InstanceType" : {
      "Description" : "Type of EC2 instance to launch",
      "Type" : "String",
      "Default" : "m1.small"
    },
    "WebServerPort" : {
      "Description" : "TCP/IP port of the web server",
      "Type" : "String",
      "Default" : "8888"
    },
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "64" },
      "m1.small"    : { "Arch" : "32" },
      "m1.large"    : { "Arch" : "64" },
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "32" },
      "c1.xlarge"   : { "Arch" : "64" },
      "cc1.4xlarge" : { "Arch" : "64" }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
    }
  },

  "Resources" : {
    "WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "2",
        "MaxSize" : "2",
        "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
      }
    },

    "LaunchConfig" : {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" },
                                          "Arch" ] } ] },
        "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }},
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "InstanceType" : { "Ref" : "InstanceType" }
      }
    },

    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : { "Ref" : "WebServerPort" },
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort" }, "/"]]},
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access and HTTP access on the inbound port",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        },
        {
          "IpProtocol" : "tcp",
          "FromPort" : { "Ref" : "WebServerPort" },
          "ToPort" : { "Ref" : "WebServerPort" },
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  },

  "Outputs" : {
    "URL" : {
      "Description" : "URL of the website",
      "Value" :  { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]}
    }
  }
}
```

Parameters

Mappings

Resources

Outputs

```
"Parameters" : {
  "InstanceType" : {
    "Description" : "Type of EC2 instance to launch",
    "Type" : "String",
    "Default" : "m1.small"
  },
  "WebServerPort" : {
    "Description" : "TCP/IP port of the web server",
    "Type" : "String",
    "Default" : "8888"
  },
  "DatabaseName": {
    "Default": "SampleDatabase",
    "Description" : "Name of the sample database",
    "Type": "String"
  },
  "DatabaseUser": {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "Sample database admin account username",
    "Type": "String"
  },
  "DatabasePwd": {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "Sample database admin account password",
    "Type": "String"
  },
  "DatabasePort": {
    "Default": "8443",
    "Description" : "TCP/IP port for the RDS database",
    "Type": "String"
  },
  "KeyName" : {
    "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
    "Type" : "String"
  }
},
```

```
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"    : { "Arch" : "64" },
    "m1.small"    : { "Arch" : "32" },
    "m1.large"    : { "Arch" : "64" },
    "m1.xlarge"   : { "Arch" : "64" },
    "m2.xlarge"   : { "Arch" : "64" },
    "m2.2xlarge"  : { "Arch" : "64" },
    "m2.4xlarge"  : { "Arch" : "64" },
    "c1.medium"   : { "Arch" : "32" },
    "c1.xlarge"   : { "Arch" : "64" },
    "cc1.4xlarge" : { "Arch" : "64" }
  },
  "AWSRegionArch2AMI" : {
    "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
    "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
    "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
    "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
    "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
  }
},
```

```
"Resources" : {
  "WebServerGroup" : {
    "Type" : "AWS::AutoScaling::AutoScalingGroup",
    "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "3",
      "MaxSize" : "3",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
    }
  },
```

```
    "SampleDatabase": {
      "Properties": {
        "Engine": "MySQL5.1",
        "DBName": {
          "Ref": "RailDatabaseName"
        },
        "Port": "8443",
        "MultiAZ" : { "Fn::FindInMap" : [ "AWSRegionCapabilities",
{ "Ref" : "AWS::Region" }, "RDSMultiAZ"] },
        "MasterUsername": {
          "Ref": "DatabaseUser"
        },
        "DBInstanceClass": "db.m1.small",
        "DBSecurityGroups": [
          {
            "Ref": "DBSecurityGroup"
          }
        ],
        "AllocatedStorage": "5",
        "MasterUserPassword": {
          "Ref": "DatabasePwd"
        }
      },
      "Type": "AWS::RDS::DBInstance"
    },
```

```
    "LaunchConfig" : {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" :
[ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
{ "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
"InstanceType" },
                                                "Arch" ] } ] },
        "SecurityGroups" : [ { "Ref" :
"InstanceSecurityGroup" } ],
        "InstanceType" : { "Ref" : "InstanceType" }
      }
    },
```

```
"UserData": {
          "Fn::Base64": {
            "Fn::Join": [
              ":",
              [
                {
                  "Ref": "DatabaseName"
                },
                {
                  "Ref": "DatabaseUser"
                },
                {
                  "Ref": "DatabasePwd"
                },
                {
                  "Ref": "DatabasePort"
                },
                {
                  "Fn::GetAtt": [
                    "SampleDatabase",
                    "Endpoint.Address"
                  ]
                },
                {
                  "Ref": "WebServerPort"
                }
              ]
            ]
          }
```

```
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : { "Ref" : "WebServerPort" },
      "Protocol" : "HTTP"
    } ],
        "HealthCheck" : {
          "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" :
"WebServerPort" }, "/"]]},
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
    }
  },
```

```
"DBSecurityGroup": {
  "Properties": {
    "DBSecurityGroupIngress": {
      "EC2SecurityGroupName": {
        "Ref": "EC2SecurityGroup"
      }
    },
    "GroupDescription": "database access"
  },
  "Type": "AWS::RDS::DBSecurityGroup"
},

"InstanceSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
      "GroupDescription" : "Enable SSH access and HTTP access on the inbound port",
      "SecurityGroupIngress" : [ {
        "IpProtocol" : "tcp",
        "FromPort" : "22",
        "ToPort" : "22",
        "CidrIp" : "0.0.0.0/0"
      },
      {
        "IpProtocol" : "tcp",
        "FromPort" : { "Ref" : "WebServerPort" },
        "ToPort" : { "Ref" : "WebServerPort" },
        "CidrIp" : "0.0.0.0/0"
      } ]
    }
  }
},
```

```
"Outputs" : {
  "URL" : {
    "Description" : "URL of the website",
    "Value" :  { "Fn::Join" : [ "", [ "http://",
{ "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]}
    }
  }
```

Create stack

AWS Management Console

https://console.aws.amazon.com/cloudformation/home?region=us-west-1

aws.amazon.com | AWS | Products | Developers | Community | Support | Account | **Welcome, Matt Wood** | Settings | Sign Out

AWS Elastic Beanstalk | Amazon S3 | Amazon EC2 | Amazon VPC | Amazon CloudWatch | Amazon Elastic MapReduce | Amazon CloudFront | AWS CloudFormation | Amazon RDS | Amazon SNS

Stacks

Region: US West (N. California) | Create New Stack | Delete Stack | Show/Hide | Refresh

Viewing: All Stacks

You do not have any stacks. Click the button below to create a new AWS CloudFormation stack. You can use AWS CloudFormation to quickly and easily deploy your AWS infrastructure resources using one of the sample templates or your own templates. (learn more)

Create New Stack

Create Stack

Cancel

SELECT TEMPLATE — SPECIFY PARAMETERS — REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of you own templates stored in S3 or on your local hard drive.

Stack Name:

ApplicationStack

Stack Template Source:

- ○ **Use a sample template**
- ◉ **Upload a Template File**

  Choose File No file chosen
- ○ **Provide a Template URL**

☐ **Show Advanced Options**

Continue

Create Stack

Cancel

SELECT TEMPLATE — SPECIFY PARAMETERS — REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of you own templates stored in S3 or on your local hard drive.

Stack Name:

ApplicationStack

Stack Template Source:

- ○ **Use a sample template**
- ○ **Upload a Template File**
- ◉ **Provide a Template URL**

http://cloudformation-templates-eu-west-1.s3-website-eu-

☐ **Show Advanced Options**

Continue

https://console.aws.amazon.com/cloudformation/home?region=us-west-1

mazon.com | AWS | Products | Developers | Community | Support | Account | Welcome, Matt Wood | Settings

anstalk | Amazon S3 | Amazon EC

US West (N. Calif | Show/Hide

All Stacks

You do not hav ... oudFormation to quickly and eas ... tes. (learn more)

Create Stack

Cancel

SELECT TEMPLATE — SPECIFY PARAMETERS — REVIEW

Template Description: Ruby on Rails Hello World Example. This template creates an Auto Scaled sample Ruby on Rails application connected to an Amazon Relational Database Service database instance. Once created, the application displays the server information. The appliction is behind an Elastic Load Balancer with cookie stickiness enabled.

Specify Parameters

Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

DatabasePort

8443

...DS database

OperatorEmail matthew@amazon.com

Email address to notify if there are any operational issues

DatabaseUser

•••••

...in account username

DatabaseName

SampleDatabase

...atabase

WebServerPort 8888

TCP/IP port for the web servers

InstanceType m1.small

EC2 instance type for the web servers

KeyName

Name of an existing EC2 KeyPair to enable SSH access to the web servers

‹ Back | Continue

...8 - 2011, Amazon W ... An amazon.com

AWS Management Console

https://console.aws.amazon.com/cloudformation/home?region=us-west-1

aws.amazon.com | AWS | Products | Developers | Community | Support | Account | Welcome, Matt Wood | Settings | Sign Out

AWS Elastic Beanstalk | Amazon S3 | Amazon EC2 | Amazon VPC | Amazon CloudWatch | Amazon Elastic MapReduce | Amazon CloudFront | AWS CloudFormation | Amazon RDS | Amazon SNS

Stacks

Region: US West (N. California) | Create New Stack | Delete Stack | Show/Hide | Refresh

Viewing: All Stacks | 1 to 1 of 1 Items

Stack Name	Created	Status	Description
ApplicationStack	2011-04-01T08:19:03Z	CREATE_IN_PROGRESS	Example application

0 Stacks selected

Select a stack above

1 Stack selected

Stack: ApplicationStack

Description | Outputs | Resources | **Events** | Template | Parameters

Stack Events

Time	Type	Logical ID	Status	Reaso
2011-04-01 09:19 GMT+0100	AWS::RDS::DBInstance	SampleDatabase	CREATE_IN_PROGRESS	
2011-04-01 09:19 GMT+0100	AWS::RDS::DBSecurityGroup	DBSecurityGroup	CREATE_COMPLETE	
2011-04-01 09:19 GMT+0100	AWS::RDS::DBSecurityGroup	DBSecurityGroup	CREATE_IN_PROGRESS	
2011-04-01 09:19 GMT+0100	AWS::ElasticLoadBalancing::LoadBalancer	ElasticLoadBalancer	CREATE_COMPLETE	
2011-04-01 09:19 GMT+0100	AWS::ElasticLoadBalancing::LoadBalancer	ElasticLoadBalancer	CREATE_IN_PROGRESS	

AWS Management Console

https://console.aws.amazon.com/cloudformation/home?region=eu-west-1

aws.amazon.com | AWS | Products | Developers | Community | Support | Account | Welcome, Matt Wood | Settings | Sign Out

AWS Elastic Beanstalk | Amazon S3 | Amazon EC2 | Amazon VPC | Amazon CloudWatch | Amazon Elastic MapReduce | Amazon CloudFront | AWS CloudFormation | Amazon RDS | Amazon SNS

Stacks

Region: EU West (Ireland) | Create New Stack | Delete Stack | Show/Hide | Refresh

Viewing: All Stacks | 1 to 1 of 1 Items

Stack Name	Created	Status	Description
ApplicationStack	03-31T20:44:14Z	CREATE_COMPLETE	Example application

1 Stack selected

Stack: Stacky

Description | **Outputs** | Resources | Events | Template | Parameters

Stack Outputs

Output values may have been specified by the template author and will be available when stack creation is complete.

Refresh

Key	Description	Value
ELB URL	URL of website	165783690.eu-.west-1.elb.

 | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com company

Steady state

monitoring with CloudWatch

Update

with CloudFormation

Update

with Puppet

Define manifest

Resource lists, dependencies

```
define apache::site ( $ensure = 'present', $require_package
= 'apache', $content = '', $source = '') {
   include apache

   $site_file = "${module_dir_path}/apache/sites/${name}"
   config_file {
      $site_file:
         ensure => $ensure,
         content => $content,
         source => $source,
         notify => Exec["reload-apache"]
   }
}
```

```
define apache::site ( $ensure = 'present', $require_package
= 'apache', $content = '', $source = '') {
    include apache

    $site_file = "${module_dir_path}/apache/sites/${name}"
    config_file {
        $site_file:
            ensure => $ensure,
            content => $content,
            source => $source,
            notify => Exec["reload-apache"]
    }
}
```

```
define apache::site ( $ensure = 'present', $require_package
= 'apache', $content = '', $source = '') {
   include apache

   $site_file = "${module_dir_path}/apache/sites/${name}"
   config_file {
      $site_file:
         ensure => $ensure,
         content => $content,
         source => $source,
         notify => Exec["reload-apache"]
   }
}
```

Apply manifest

`puppet apply`,
Pull/push from the Puppet Master

Performance automation

with EC2 autoscaling

```
as-create-launch-config
      AppLaunchConfig
      --image-id ami-13221667
      --instance-type m1.large
      --key amazon-web
      --group "Web and SSH"
```

```
as-create-auto-scaling-group
 AppScalingGroup
 --launch-configuration AppLaunchConfig
--availability-zones eu-west-1a, eu-west-1b
--min-size 10
--max-size 100
--load-balancers app-load-balancer
```

```
as-put-scaling-policy
 AppScaleUpPolicy
 --auto-scaling-group AppScalingGroup
--scaling-adjustment 1
--type ChangeInCapacity
--cool-down 300
```

```
mon-put-metric-alarm
 AppHighCPUAlarm
 --comparison-operator GreaterThanThreshold
 --evaluation-period 1
 --metric-name CPUUtilization
 --namespace “AWS:EC2”
--period 600
--statistic Average
--threshold 80
--alarm-actions <high-cpu-policy-arn>
--dimensions
“AutoscalingGroupName=AppScalingGroup”
```

```
as-put-scaling-policy
 AppScaleDownPolicy
 --auto-scaling-group AppScalingGroup
--scaling-adjustment -1
--type ChangeInCapacity
--cool-down 300
```

```
mon-put-metric-alarm
 AppLowCPUAlarm
 --comparison-operator LessThanThreshold
 --evaluation-period 1
 --metric-name CPUUtilization
 --namespace “AWS:EC2”
--period 600
--statistic Average
--threshold 80
--alarm-actions <low-cpu-policy-arn>
--dimensions
“AutoscalingGroupName=AppScalingGroup”
```

aws.amazon.com/cloudformation

---

puppetlabs.com

opscode.com/chef

---

aws.amazon.com/whitepapers

AGENDA

Orchestrating the Cloud

- ☐ 1. Application architecture
- ☐ 2. Role of orchestration
- ☐ 3. Pillars of orchestration
- ☐ 4. Orchestration by example
- ☐ 5. Summary

3 tiers of cloud application design

Maximising the value in each tier

Orchestration codifies knowledge

Three pillars of orchestration

Provisioning orchestration

Configuration management

Performance automation

CloudFormation

Puppet, Chef

Autoscaling service

aws.amazon.com

Thank you!

QUESTIONS + COMMENTS

matthew@amazon.com

@mza

ON TWITTER

amazon web services™