SOFTPROM

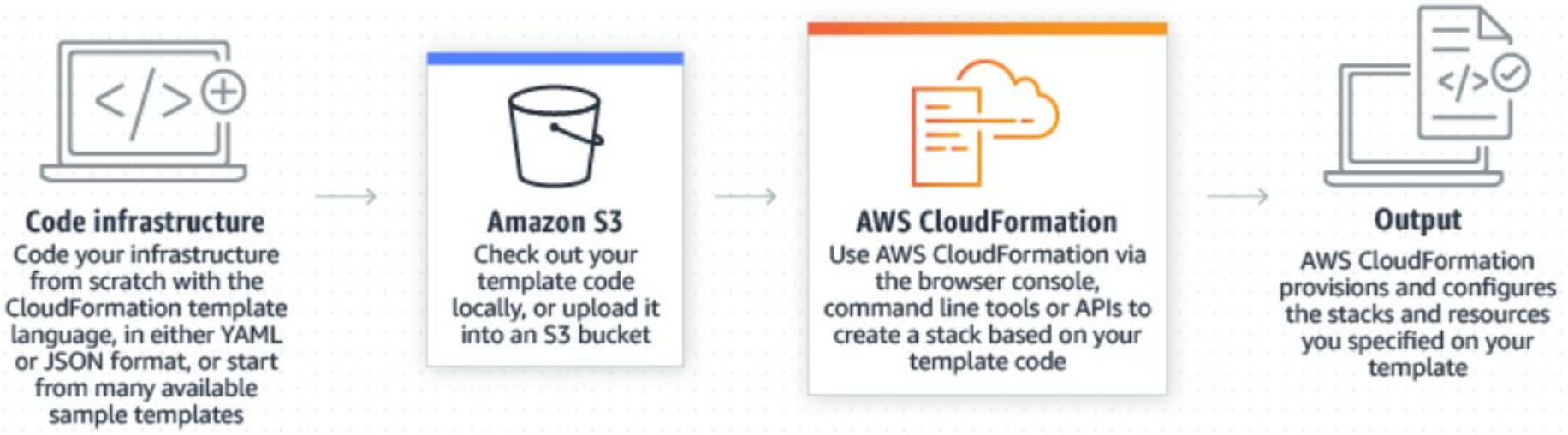
Do awesome things with Infrastructure as Code

Vladimir Grigorenko - Cloud Solutions Architect

Why infrastructure as a code?

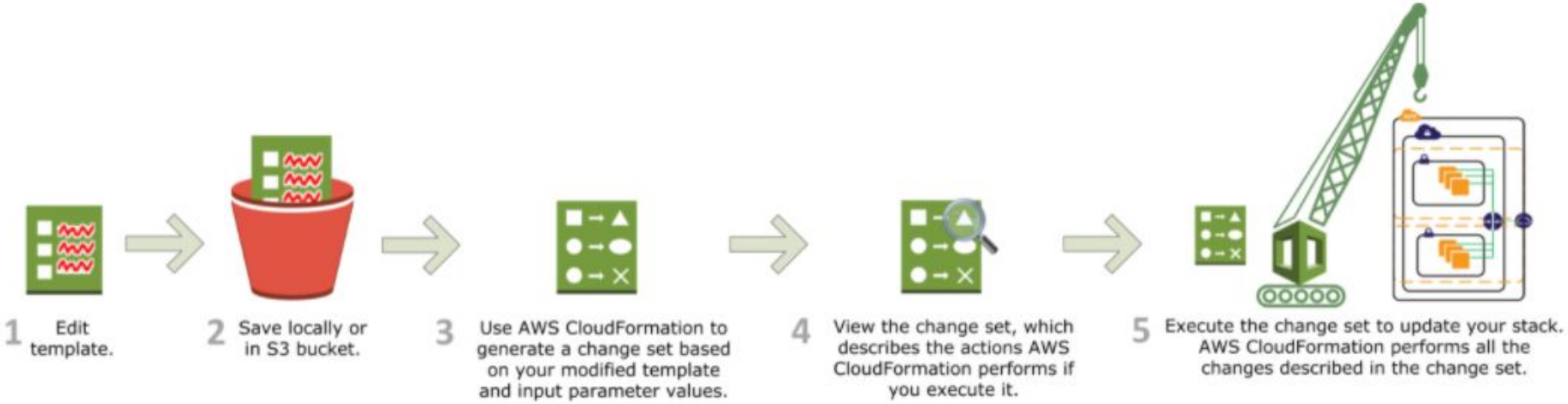
- Одни и те же действия при настройке. Рутина.
- Человеческий фактор.
- Написание документации.
- Prod/Stage/Dev инфраструктура.
- Модернизация инфраструктуры, версионность.
- Распространение готовых решений для широкого круга заказчиков.

How does AWS CloudFormation work?



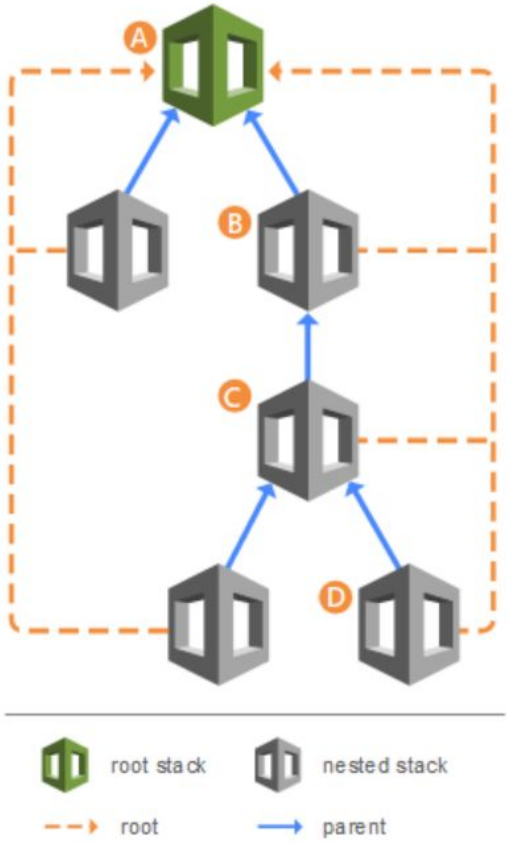
amazon web services Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Updating stacks using change sets



amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Nested Stacks



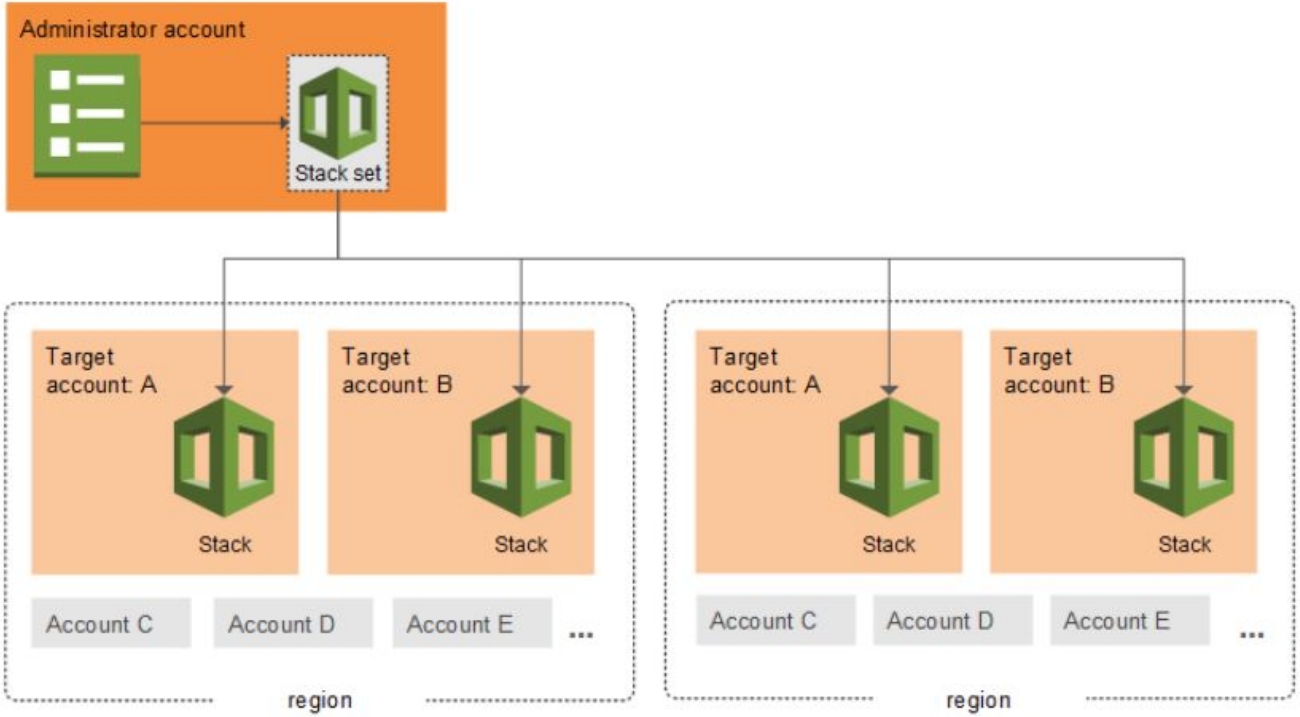

```yaml
Resources:

  # Authentication
  Authentication:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://s3.amazonaws.com/${ResourceBucket}/authentication.yaml
      Parameters:
        AuthName: !Sub ${ParentStackName}-${AWS::Region}
        PinpointAppId: !GetAtt Pinpoint.Outputs.PinpointAppId

  # Buckets
  Buckets:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://s3.amazonaws.com/${ResourceBucket}/base/buckets.yaml
      Parameters:
        CleanupBucketLambdaArn: !Ref CleanupBucketLambdaArn
```

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

AWS CloudFormation StackSets



amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

AWSTemplateFormatVersion: "version date"

Description:
String

Metadata:
template metadata

Parameters:
set of parameters

Rules:
set of rules

Mappings:
set of mappings

Conditions:
set of conditions

Transform:
set of transforms

Resources:
set of resources

Outputs:
set of outputs

Format Version	"AWSTemplateFormatVersion" : "2010-09-09"
Description	"Description" : "Here are some details about the template."

Metadata	`Metadata:`
	`  Instances:`
	`  Description: "Information about the instances"`
	`  Databases:`
	`  Description: "Information about the databases"`
	
	• AWS::CloudFormation::Init
	`  commands:`
	`  test:`
	`  command: "echo \"$MAGIC\" > test.txt"`
	`  env:`
	`  MAGIC: "I come from the environment!"`
	`  cwd: "~"`
	• AWS::CloudFormation::Interface
	`  ParameterGroups:`
	`  - ParameterGroup`
	`  ParameterLabels:`
	`  ParameterLabel`
	• AWS::CloudFormation::Designer

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Metadata	Type: AWS::CloudFormation::Authentication
	String:
	  accessKeyId: String
	  buckets:
	  - String
	  password: String
	  secretKey: String
	  type: String
	  uris:
	  - String
	  username: String
	  roleName: String

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Parameters	

```
Parameters:
  InstanceTypeParameter:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
    Description: Enter t2.micro, m1.small, or m1.large.
Default is t2.micro.

Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType:
        Ref: InstanceTypeParameter
      ImageId: ami-0ff8a91507f77f867
```

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Rules

```
Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':
            - - a1.medium
              - !Ref InstanceType
        AssertDescription: 'For a test environment, the instance type must be
a1.medium'
  prodInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - prod
    Assertions:
      - Assert:
          'Fn::Contains':
            - - a1.large
              - !Ref InstanceType
        AssertDescription: 'For a production environment, the instance type must be
a1.large'
```

Mappings

```
Mappings:
  RegionMap:
    us-east-1:
      "HVM64": "ami-0ff8a91507f77f867"
    us-west-1:
      "HVM64": "ami-0bdb828fd58c52235"
    eu-west-1:
      "HVM64": "ami-047bb4163c506cd98"
```

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Conditions

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Description: Environment type.
    Default: test
    Type: String
    AllowedValues:
      - prod
      - test
    ConstraintDescription: must specify prod or test.
Conditions:
  CreateProdResources: !Equals
    - !Ref EnvType
    - prod
Resources:
  EC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-0ff8a91507f77f867
  MountPoint:
    Type: 'AWS::EC2::VolumeAttachment'
    Condition: CreateProdResources
    Properties:
      InstanceId: !Ref EC2Instance
      VolumeId: !Ref NewVolume
      Device: /dev/sdh
  NewVolume:
    Type: 'AWS::EC2::Volume'
    Condition: CreateProdResources
    Properties:
      Size: 100
      AvailabilityZone: !GetAtt
        - EC2Instance
```

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Transform	// Start of processable content for MyMacro and AWS::Serverless
	Transform:
	- MyMacro
	- 'AWS::Serverless'
	Resources:
	WaitCondition:
	Type: 'AWS::CloudFormation::WaitCondition'
	MyBucket:
	Type: 'AWS::S3::Bucket'
	Properties:
	BucketName: MyBucket
	Tags: [{"key":"value"}]
	CorsConfiguration:[]
	MyEc2Instance:
	Type: 'AWS::EC2::Instance'
	Properties:
	ImageID: "ami-123"
	// End of processable content for MyMacro and AWS::Serverless

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

Template anatomy

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Resources	`Resources:`
	`  MyEC2Instance:`
	`  Type: "AWS::EC2::Instance"`
	`  Properties:`
	`  ImageId: "ami-0ff8a91507f77f867"`
	`  InstanceType: t2.micro`
	`  KeyName: testkey`
	`  BlockDeviceMappings:`
	`  -`
	`  DeviceName: /dev/sdm`
	`  Ebs:`
	`  VolumeType: io1`
	`  Iops: 200`
	`  DeleteOnTermination: false`
	`  VolumeSize: 20`

Outputs	`Outputs:`
	`  Logical ID:`
	`  Description: Information about the value`
	`  Value: Value to return`
	`  Export:`
	`  Name: Value to export`

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER
SOFTPROM
softprom.com • info@softprom.com

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER

SOFTPROM
softprom.com • info@softprom.com

Benefits

Model it all	AWS CloudFormation allows you to model your entire infrastructure and application resources with either a text file or programming languages.
Automate & deploy	AWS CloudFormation provisions your application resources in a safe, repeatable manner, allowing you to build and rebuild your infrastructure and applications, without having to perform manual actions or write custom scripts.
It's just code	Codifying your infrastructure allows you to treat your infrastructure as just code.

- An easy way to create & manage a collection of AWS resources
- Allows orderly and predictable provisioning and updating of resources
- Allows you to version control your AWS infrastructure
- Deploy and update stacks using console, command line or API
- You only pay for the resources you create

Softprom - Advanced Consulting Partner в сети Amazon Web Services.

softprom.com | aws@softprom.com