

# FirstNet Web Octopus Pipelines

The objective is to explain FirstNet Web tier deployment and release pipelines. The following three pipelines are being covered:

- FirstNet-Web deployment pipeline (Including DR)
  - FirstNet-Web Rollback
  - SOE Image customisation pipeline
  - FirstNet-Web BAU pipeline
- 
- [Infra as Code \(IaC\) using Bicep](#)
  - [Infra Resources](#)
  - [Load Balancing, Healthcheck, and Auto scaling](#)
    - [Load Balancer Rules](#)
    - [Healthcheck Probs](#)
    - [Auto Scaling](#)
  - [FirstNet-Web deployment pipeline](#)
    - [Overview](#)
    - [Prerequisites](#)
    - [Variables](#)
    - [Steps](#)
  - [DR Deployment](#)
    - [Overview](#)
  - [Rollback Pipeline](#)
    - [Overview](#)
    - [Variables](#)
    - [Steps](#)
  - [FirstNet SOE Image Pipeline](#)
    - [Overview](#)
    - [Steps](#)
  - [BAU Pipeline](#)
    - [Getting Current Deployment Info](#)
    - [Manual Scale Up-Down](#)
    - [Switch from DR to Primary - Only Current](#)
  - [Process for Release versioning](#)

## Infra as Code (IaC) using Bicep

Firstnet-Web tier is fully sitting on Microsoft Azure and using cloud infrastructure and networking. As part of deployment pipeline, it is designed to provision the entire infrastructure upon every release and deploy the application on the new infrastructure. This will enable a clean Blue/Green deployment and a safe rollback approach as the old instances are available and running after deployment.

The code for infrastructure is written using Microsoft Bicep and stored in a git repository ([ColonialFirstState/firstnet.infrastructure: This is part of ODP migration into new Azure environment \(internal.cba\)](#)).

There is also a teamcity ([firstnet.infrastructure — TeamCity \(internal.cba\)](#)) pipeline which builds Bicep, generate ARM templates, create nuget package and upload it to the artifactory. So, everytime there is a change in the template, the teamcity pipeline must be triggered to generate and publish nuget package which is being used later in deployment pipeline.

## Infra Resources

The following Azure resources are used for firstnet web tier:

- **Azure Virtual Machine Scaleset (VMSS)**  
This service provides a set of load balanced virtual machines and makes sure the given number are always up and healthy. Autoscaling policies are applied to the VMSS.
- **Azure Load Balancer (Public and Internal)**  
Internal Load Balancer in non-prod and public Load Balancer in prod environment are used in front of VMSS. Healthcheck probs are added to the LB to ensure healthy VM instances of the VMSS.
- **Azure Public IP Address**  
A public IP address is being assigned to the public LB. This IP will be configured in DNS service (Route53) to route traffic to the services.
- **Azure Storage Account**  
Azure blob storage is used to keep release packages. The deployment pipeline pushes the application packages into this blob and later when provisioning the VMs pulls them from this storage and installs them on the VMs.

## Load Balancing, Healthcheck, and Auto scaling

## Load Balancer Rules

Traffic to tcp ports 80 and 443 are allowed to pass load balancer. The following is how these rules are added to load balancer in the template:

```
{
  name: '${lbRuleName}-443'
  properties: {
    backendAddressPool:{
      id: resourceCid('Microsoft.Network/loadBalancers/backendAddressPools', lbName, lbBackendPoolName)
    }
    backendPort: 443
    enableFloatingIP: false
    frontendIPConfiguration: {
      id: resourceCid('Microsoft.Network/loadBalancers/frontendIPConfigurations', lbName, lbFrontend)
    }
    frontendPort: 443
    probe:{
      id: resourceCid('Microsoft.Network/loadBalancers/probes', lbName, lbProbName)
    }
    protocol: 'Tcp'
    idleTimeoutInMinutes: 30
    loadDistribution: 'Default'
  }
}
{
  name: '${lbRuleName}-80'
  properties: {
    backendAddressPool:{
      id: resourceCid('Microsoft.Network/loadBalancers/backendAddressPools', lbName, lbBackendPoolName)
    }
    backendPort: 80
    enableFloatingIP: false
    frontendIPConfiguration: {
      id: resourceCid('Microsoft.Network/loadBalancers/frontendIPConfigurations', lbName, lbFrontend)
    }
    frontendPort: 80
    probe:{
      id: resourceCid('Microsoft.Network/loadBalancers/probes', lbName, lbProbName)
    }
    protocol: 'Tcp'
    idleTimeoutInMinutes: 30
    loadDistribution: 'Default'
  }
}
```

## Healthcheck Probs

The following health prob is set for the Load Balancer:

```
{
  protocol: 'Http'
  port: 8080
  intervalInSeconds: 10
  numberOfProbes: 3
  requestPath: '/health/status'
}
```


This is a http healthcheck on port 80 and it is against a generic health check endpoint deployed on each boxes. Any service on the box not being healthy will cause this endpoint to return false.

## Auto Scaling

currently there are two types of auto scaling configured for the VMSS. Schedule based and Metric based. The following is the configuration for auto scaling: (this configuration can be find/updated under the autoscaling settings in the template)


Week Day:

Default policy, 4 instances (staging) and 8 instances (Production) during the week days.

Week Day 

Scale mode

☐ Scale based on a metric ☒ Scale to a specific instance count

Instance count <sup>\*</sup> 

4

Schedule

☐ Specify start/end dates ☒ Repeat specific days

Repeat every

☒ Monday

☒ Tuesday

☒ Wednesday


☒ Thursday

☒ Friday

☐ Saturday

☐ Sunday

Timezone

(UTC+10:00) Canberra, Melbourne, Sydney 

Start time

07:00


End time

Enter end time in 24 hour format

Specify an end time, else this scale condition will apply for all days until it reaches the start time of another scale condition


Week Night:

Scales down to half, 2 instances (staging) and 4 instances (production) during the night.

Week Night 

Scale mode

☐ Scale based on a metric ☒ Scale to a specific instance count

Instance count <sup>\*</sup> 

2

Schedule

☐ Specify start/end dates ☒ Repeat specific days

Repeat every

☒ Monday

☒ Tuesday

☒ Wednesday


☒ Thursday

☒ Friday

☐ Saturday

☐ Sunday

Timezone

(UTC+10:00) Canberra, Melbourne, Sydney 

Start time

01:00

End time

Enter end time in 24 hour format

Specify an end time, else this scale condition will apply for all days until it reaches the start time of another scale condition

Weekend:

Scales down to half, 2 instances (staging) and 4 instances (production) during the weekend.

Weekend

Scale mode

☐ Scale based on a metric
☒ Scale to a specific instance count

Instance count

2

Schedule

☐ Specify start/end dates
☒ Repeat specific days

Repeat every

☐ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday

☐ Friday
☒ Saturday
☒ Sunday

Timezone

(UTC+10:00) Canberra, Melbourne, Sydney

Start time

01:00

End time

Enter end time in 24 hour format

Specify an end time, else this scale condition will apply for all days until it reaches the start time of another scale condition

CPU Metric:

Scales up/down based on CPU percentage usage in the instances.

CPU Metric

Scale mode

☒ Scale based on a metric
☐ Scale to a specific instance count

Rules

Scale out

Whenfnwebstg-vmss-bc21b... (Average) Percentage CPU > 70Increase count by 2

Scale in

Whenfnwebstg-vmss-bc21b... (Average) Percentage CPU < 40Decrease count by 1

+ Add a rule

Instance limits

Minimum

2

Maximum

4

Default

2

Schedule

☐ Specify start/end dates
☒ Repeat specific days

Repeat every

☒ Monday
☒ Tuesday
☒ Wednesday
☒ Thursday

☒ Friday
☐ Saturday
☐ Sunday

Timezone

(UTC+10:00) Canberra, Melbourne, Sydney

Start time

07:00

End time

Enter end time in 24 hour format

Specify an end time, else this scale condition will apply for all days until it reaches the start time of another scale condition

## FirstNet-Web deployment pipeline

### Overview

This pipeline (aka integration pipeline) [Overview - Octopus Deploy \(internal.cba\)](#) is automating the process of provisioning a brand new environment "next" and having a full web-tier installed on it. After testing and verifying "next", it allows to swap "next" and "current" environments and route the traffic into new environment. The followings tech stacks are utilized to implement this pipeline:

- Infrastructure as Code (IaC) - Microsoft Bicep is used to code the infrastructure. All the infra codes and pipeline scripts are in the firstnet. infrastructure git repository (<https://github.source.internal.cba/ColonialFirstState/firstnet.infrastructure.git>)
- Microsoft Azure Resource Group and Tagging features are used to implement Blue/Green release strategy. There resource groups are used to isolate resources for different stages (Current, Next, Previous, Storage)
- AWS Route53 DNS routing is used to control the traffic flow.

## Prerequisites

The pipeline requires 4 resource group per environment to be allocated for that environment. This is through tagging the resource group ( `fnEnvironment=EnvironmentName` ). This tagging needs to be performed manually as part of provisioning a new environment process.

For instance, to make the staging environment ready for the pipeline, the "fnEnvironment" tag must be set to "FirstNet-Web-STG" (Octopus environment name for staging) for the following resource groups:

Primary (australiaeast)	DR (australiasoutheast)
syd-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-01-rg	mel-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-01-rg
syd-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-02-rg	mel-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-02-rg
syd-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-03-rg	mel-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-03-rg
syd-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-04-rg	mel-cfs-firstnet-ga-nonp-v0-t4-stg-fnweb-04-rg

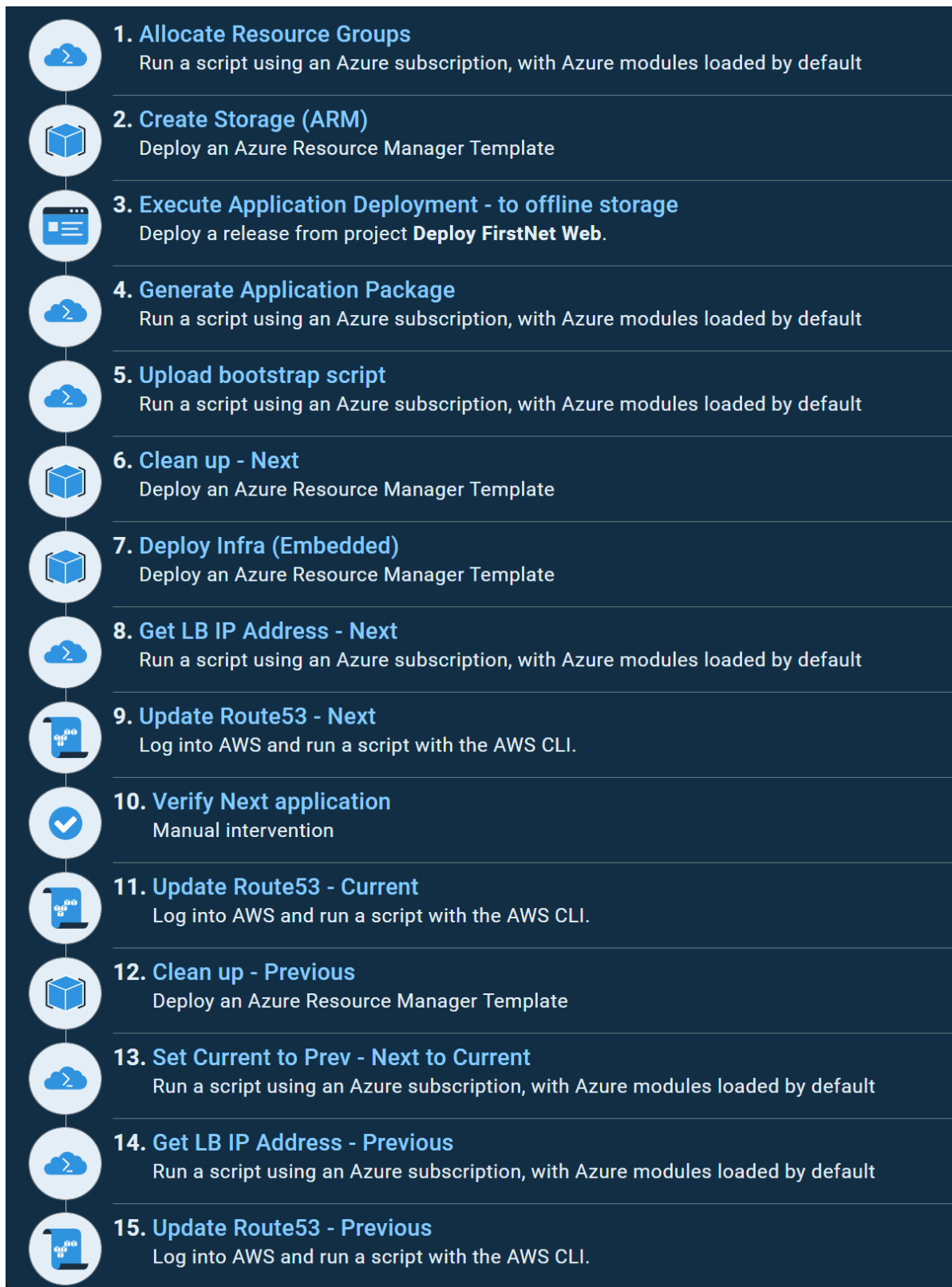
## Variables

Note: There are some variables which are being set in the pipeline dynamically and doesnt need any initial value. Below only lists the variables which are required to be set:

Name	Description
AdminUsername	Credentials used to provision the VMs with. There will be a user to login to the VMs.
AdminPassword	
API Key	
Azure Account	Azure SP account for CLI and template execution accesses to azure resources
AzureImageId	VM Image to create VMs from. This must be updated every time the base image is updated.
AzureSubnet	The subnet in which resources will be deployed to.
AzureVnet	The Vnet in which resources will be deployed. It is different between regions so the value has a switch condition based on the region.
CleanupTemplatePath	Path to the template used to clean up a resource group. This template is in the infra code package.
CurrentDnsPrefix	DNS subdomain for 'current' environment. This value is different per environment (ex: <a href="#">www.t1</a> , <a href="#">www.stg</a> , etc)
NamingPrefix	Naming prefix used to name azure resources.
NextDnsPrefix	DNS subdomain for 'next' environment. This value is different per environment (ex: next. <a href="#">www.t1</a> , next. <a href="#">www.stg</a> , etc )
PackagePassword	
ParamFilePath	Path to the parameter file for templates. This file is in infra code package.
PreviousDnsPrefix	DNS subdomain for 'previous' environment. This value is different per environment (ex: previous. <a href="#">www.t1</a> , previous. <a href="#">www.stg</a> , etc)
Region	The region which pipeline will be run against. This is a prompted variable and is asked every time pipeline is ran. For DR deployment the DR region needs to be set.
Route53HostedZone	Hosted zone for deployment environment. (ex: <a href="#">03M3.s6.privateonline.com.au</a> )
StorageParamPath	Path to the parameter file for storage template. This file is in infra code package.
StorageSku	sku for storage account
StorageTemplatePath	Path to the template used to create storage account.

TemplateFilePath	Path to the template files used to create infra resources in azure.
VmssZones	Azure availability zones for VMSS deployment. This value is being switched based on the region due to no AZ in one of azure regions.

## Steps



## 1 - Allocate Resource Group

Getting allocated resource groups for the deployment environment (where `fnEnvironment` is equal to Environment Name) and identify the stage for each resource group (Current, Next, Previous, Storage). Also the "next" resource group is being allocated as part of this step by tagging a free resource group. "next" resource group will be used to deploy the new infrastructure later in the pipeline.

## **2 - Create Storage**

Azure Storage account is used to keep application release packages and scripts being used to install and configure the applications on the VMs. This step creates the storage account only if it does not exist. The storage account will be created in the "storage" resource group tagged by Step 1.

## **3 - Execute Application Deployment - to offline storage**

This step runs a child pipeline which is responsible to build application packages and upload them into offline storage.

## **4 - Generate Application Package**

This step downloads application packages from artifactory, packages them and uploads them to azure storage account created in step 2. Also deployment scripts being used to install applications on the VMs later are uploaded into azure storage account at this step.

## **5 - Upload bootstrap script**

Bootstrap script contains the logic to trigger installing and configuring applications on the VM. The bootstrap script is executed in the VMSS extension later in the pipeline. This step is uploading the bootstrap script to azure storage account.

## **6 - Clean up "next"**

This step runs clean up script against "next" resource group (identified by step 1 earlier in the pipeline) to make sure the new infrastructure will be deployed in a clean surface.

## **7 - Deploy Infra**

This step runs the templates against "next" resource group (identified by step 1 earlier in the pipeline) to provision the infrastructure. As part of the template execution a VMSS extension will be setup to download and execute bootstrap script from the storage account. The bootstrap script then will handle the logic to install and configure the applications on the VM. By the end of this step there will be a brand new infrastructure with all the applications install on it in the "next" resource group.

## **8 - Get LB IP Address for "next"**

This step queries the load balancer in the "next" resource group and gets its IP address. This IP address will be used to update DNS later in the pipeline.

## **9 - Update Route53 for "next"**

This step updates Route53 and sets DNS record for "next" dns to the load balancer IP obtained in step 8.

## **10 - Verify "Next" Application**

At this step the "next" stack is up and running and we have the "next" url pointing to it. This step pauses the pipeline. After testing "next" stack and verifying the deployment manually this step requires a manual intervention to proceed.

## **11 - Update Route53 Current**

This step updates Route53 and sets DNS record for "current" to the load balancer IP obtained in step 8.

## **12 - Cleanup Previous**

This step only executes if all previous steps are successful and there is an old "previous" instance running. (the previous of old current) It removes the instances and cleans up the resource group.

## **13 - Set Current to Prev - Next to Current**



This step swaps resource groups by updating their tags. It sets the "current" to "previous" and "next" to "current"

#### 14 - Get LB IP Address - Previous

This step queries the load balancer in the "previous" resource group (old current) and gets its IP address. This IP address will be used to update DNS later in the pipeline.

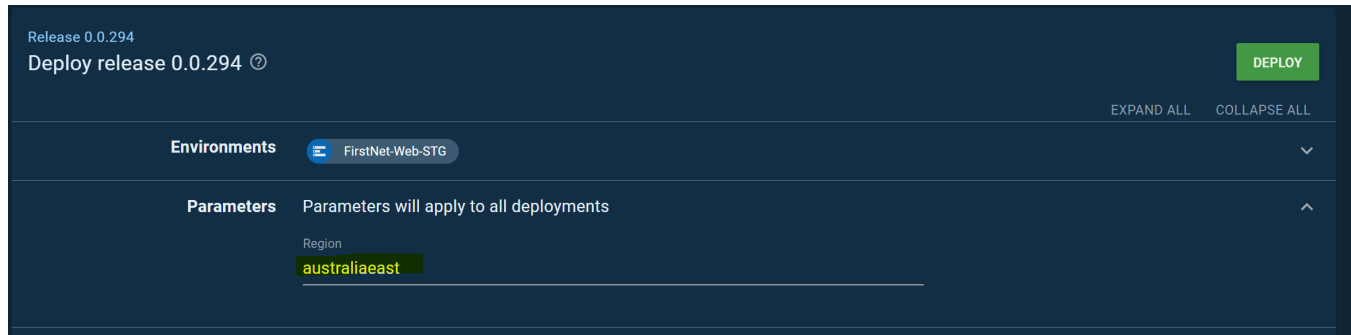
#### 15 - Update Route53 - Previous

This step updates Route53 and sets DNS record for "previous" to the load balancer IP obtained in step 14.

## DR Deployment

### Overview

Same Web-deployment pipeline can be used to deploy into DR site (australiasoutheast - Melbourne region). Region is an octopus variable with default value to primary site (australiaeast) and is being prompted as the last step before pipeline execution.



Release 0.0.294  
Deploy release 0.0.294 ?

EXPAND ALL COLLAPSE ALL

**Environments** FirstNet-Web-STG

**Parameters** Parameters will apply to all deployments

Region  
australiaeast

Setting the region to 'australiasoutheast' will deploy into DR. Please note, the same resource group allocation rules is applied for DR deployment as well, meaning there must be four resource groups tagged for the environment under DR subscription.

Note: after DR deployment is finished, the route53 will be updated to point to the DR instances. This means just by triggering and continuing DR deployment the traffic will be routed to it automatically.

## Rollback Pipeline

### Overview

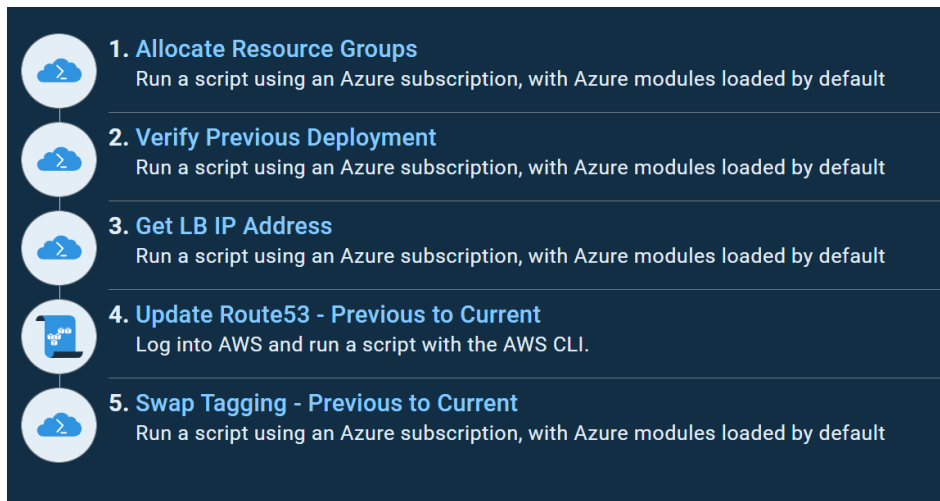
This pipeline ([Overview - Octopus Deploy \(internal.cba\)](#)) is automating the process of rolling back to the previous instance (old current) in case it is required. Please note that it can only be used to roll back from current to previous once. In case required to go back to the newer version after roll back is performed, a new deployment needs to be done using the integration pipeline.

### Variables

Name	Description
AdminUser	Credentials used to create VM. This credentials can be used to login to the VM if needed.
AdminPassword	Note: If the pipeline runs successfully the VM will be deleted by the end of pipeline, so this is only for failed run and troubleshooting.
Azure Account	The azure SP for accessing and using azure workspace.
AzureImageId	The base image ID from what the new image will be created.
AzureSubnet	Subnet in which the VM will be provisioned in.

AzureVmName	Name of the VM used to create customised image.
AzureVnet	Vnet in which the VM will be provisioned into.
ContainerName	Name of blob container used to upload/download scripts from.
ResourceGroup	The Resource Group in which the resources will be created. The VM and its dependencies will be removed by the end of pipeline. However, the created image will be stored in the same resource group.
DrResourceGroup	The created image will be copied into DR for DR use as well.
StorageAccountName	Name of storage account that is created by pipeline and used to keep script files.
StorageSku	sku of the storage account.
vmSize	Size of VM
WinHttpProxyAddress	Proxy address used by script to download updates. This value is different for prod and non-prod environments.

## Steps



### 1 - Allocate Resource Groups

This step identifies "current" and "previous" resource groups by looking at their tags.

### 2 - Verify Previous Deployment

This step verifies both "current" and "previous" are found.

### 3 - Get LB IP Address

This step obtains the LB IP address of the "previous" instance. Note that prod and non-prod environment use public and internal load balancers accordingly. Therefore, either a public IP or an internal IP is being obtained in this step.

### 4 - Update Route53 - Previous to Current

This step updates Route53 and sets the DNS to IP address obtained in step 3.

### 5 - Swap Tagging - Previous to Current

This step swaps resource groups by updating their tags. It sets the "previous" to "current" and free up the old "current" by removing its tag.

## FirstNet SOE Image Pipeline

## Overview

This pipeline ([Overview - Octopus Deploy \(internal.cba\)](#)) is used to generate FirstNet customized image using SOE image released by SOE team. It basically installed some of the required components on top of SOE image and releases it. The out put of this pipeline will be an image copied to both primary and DR subscriptions under T3 resource groups. Image below shows one of the staging T3 resource group as an example with the firstnet images created in it:

Name	Type	Location
20210817.1743.14516-fnweb-base	Image	Australia East
20210921.1254.14769-cfs-fnweb-base	Image	Australia East
20210921.1254.14769-cfs-fnweb-base_os_disk_snapshot	Snapshot	Australia East
20210921.1254.14769-cfstest-fnweb-base	Image	Australia East
20210921.1254.14769-cfstest-fnweb-base_os_disk_snapshot	Snapshot	Australia East
20210921.1254.14769-fnweb-base	Image	Australia East
20211019.1728.14917-cfs-fnweb-base	Image	Australia East
20211019.1728.14917-cfs-fnweb-base_os_disk_snapshot	Snapshot	Australia East
967c88bb-294c-4fa2-aa3b-ad8fc44f2ad1 (FirstNet_Demo)	Shared dashboard	Australia East

## Steps

- 1. Create Storage**  
Deploy an Azure Resource Manager Template
- 2. Upload bootstrap script**  
Run a script using an Azure subscription, with Azure modules loaded by default
- 3. Provision VM**  
Deploy an Azure Resource Manager Template
- 4. Check VM Job**  
Run a script using an Azure subscription, with Azure modules loaded by default
- 5. Verify DLL Version**  
Manual intervention
- 6. Create Image**  
Run a script using an Azure subscription, with Azure modules loaded by default
- 7. Cleanup**  
Run a script using an Azure subscription, with Azure modules loaded by default

### 1 - Create Storage

This step creates a storage account in T3 resource group. The storage account will be used to keep customization scripts required later in the pipeline.

### 2 - Upload bootstrap script

This step uploads the customization scripts into the storage account created in step 1.

### 3 - Provision VM

This step creates a VM using latest SOE image (configured in the pipeline variables) using a bicep template from the infra code repository. ([firstnet.infrastructure/fn-base-vm.bicep at master · ColonialFirstState/firstnet.infrastructure \(internal.cba\)](#)) as part of the template the VM uses 'VM Extension' to download scripts from storage account and execute them on the created VM. This job takes certain amount of time to complete and at the end it uploads a verification file into storage account.

#### 4 - Check VM Job

This step keeps checking the storage account for the verification file uploaded at step 3. Once found it considers the VM job is finished and it is ready to create the image from.

#### 5 - Verify DLL Version

This is a manual step to verify changes on the VM. The verification file content can be found in the logs of previous step. Once verified, the pipeline should be proceeded manually.

#### 6 - Create Image

This step creates the image from VM created and 'sysprep'ed in step 4. Once the image created it will be copied to DR T3 resource group specified in pipeline variables as well.

#### 7 - Cleanup

This step removes the storage account, VM and its other resources used to during the pipeline.

### BAU Pipeline

This pipeline ([Overview - Octopus Deploy \(internal.cba\)](#)) is used to perform some of the pre-defined functionalities against current deployments in different environments. Currently there are four runbook designed and implemented as follows:

### Getting Current Deployment Info

This runbook queries and shows some information about given environment. The following image shows the output for

Environment: FirstNet-Web-TestINT4	Info	November 12th 2021 22:43:49
Region : australiaeast	Info	November 12th 2021 22:43:49
Identifying 'Current' Resource Group ...	Info	November 12th 2021 22:43:52
'current' resource group found: syd-cfs-firstnet-ga-nonp-v0-t4-fnweb-16-rg	Info	November 12th 2021 22:43:52
VMSS Name: fn-web-t4-vmss-559c8fa6-c23e-403d-90d0-20bc1d86b560	Info	November 12th 2021 22:43:53
release : 0.0.331	Info	November 12th 2021 22:43:53
releaseDate : Friday, 12 November 2021 9:06:17 PM	Info	November 12th 2021 22:43:53
*****	Info	November 12th 2021 22:43:53
Number of instances: 1	Info	November 12th 2021 22:43:56
Location: australiaeast	Info	November 12th 2021 22:43:56
-----	Info	November 12th 2021 22:43:56
Instance Name: fn-web-t4-vmss-559c8fa6-c23e-403d-90d0-20bc1d86b560_0	Info	November 12th 2021 22:43:56
Instance Id : /subscriptions/e3570a53-d889-4d6a-98e8-705c6e031a4e/resourceGroups/syd-cfs-firstnet-ga-nonp-v0-t4-fnweb-16-rg/providers/Microsoft.Compute/virtualMachineScaleSets/fn-web-t4-vmss-559c8fa6-c23e-403d-90d0-20bc1d86b560/virtualMachines/0	Info	November 12th 2021 22:43:56
Instance Zone: 1	Info	November 12th 2021 22:43:56
Instance Nic : /subscriptions/e3570a53-d889-4d6a-98e8-705c6e031a4e/resourceGroups/syd-cfs-firstnet-ga-nonp-v0-t4-fnweb-16-rg/providers/Microsoft.Compute/virtualMachineScaleSets/fn-web-t4-vmss-559c8fa6-c23e-403d-90d0-20bc1d86b560/virtualMachines/0/networkInterfaces/fn-web-t4-nic-559c8fa6-c23e-403d-90d0-20bc1d86b560	Info	November 12th 2021 22:43:58
Instance IP : 10.211.77.27	Verbose	November 12th 2021 22:43:58
Process C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell.exe in E:\Octopus.Tentacle.v4\externalpool-iwt.octopus.deploy.internal.cba\Work\20211112114340-300332-1940 exited with code 0	Verbose	November 12th 2021 22:43:58
Updating manifest with output variables	Verbose	November 12th 2021 22:43:58
Updating manifest with action evaluated variables	Verbose	November 12th 2021 22:43:59
Released worker External Worker p03zbp0w7617.s3.chp.cba from lease WorkerTaskLeases-31921	Verbose	November 12th 2021 22:43:59
Successfully finished Get VMSS Info on a Worker	Verbose	November 12th 2021 22:43:59

### Manual Scale Up-Down

This runbook can be used to manually scale up or down a VMSS in the given environment. After selecting the environment, the number of instances should be specified and it will be set to the VMSS. This means the auto scaling configuration will be overridden by this fixed instance count and needs to be re-enabled later if needed.

Manual Scale Up-Down
Run Snapshot YP97RHR
RUN

**Environments**
Select one or many environments

FirstNet-Web-TestINT3

Select environments

**Parameters**
Parameters will apply to all runs

Region  
australiaeast

ScaleCapacity  
4

**Packages**
1 package included, at version 1.0.128

SHOW ADVANCED

## Switch from DR to Primary - Only Current

This runbook can be used to switch the traffic from DR back to the primary for 'current' instances.

## Process for Release versioning

- Generic projects like(Base systems, SSL certificates, scripts, healthcheck) will follow incremental versioning (1.0.1, 1.0.2)
- Application specific projects(Colonial, Investor, FOL, Statements, Esetup, FNE, Adviser Dashboard, statements, FileManager, Supermatch, Online Docs) will use the application artefact versioning(based on teamcity build version control)
- Infrastructure pipeline – Incremental build version
- Deploy FirstNet web pipeline(Based on firstNet classic application project)
- Deploy FirstNet main pipeline(incremental versioning)
- Minor version will be incremented if there are updates in Octopus model alone(73.2.0.21 => 73.2.0.21.1)