This month's presenters:

Microsoft





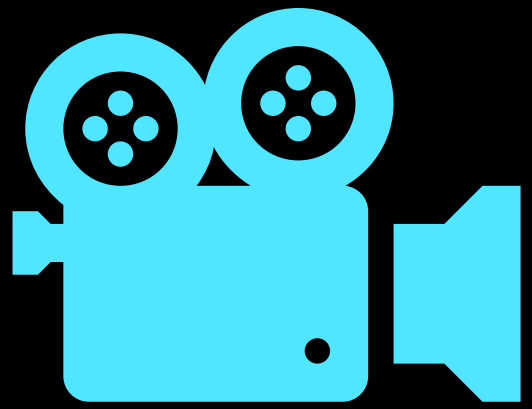
# Azure Landing Zones

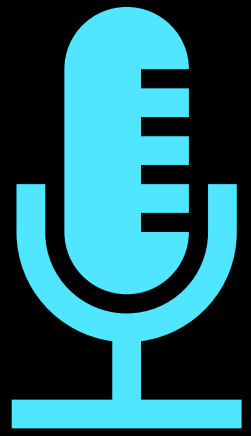
## 6th December 2023 - External Community Call

**Registration:**
https://aka.ms/ALZ/CommunityCallRegister

**Agenda (please add suggestions):**
https://aka.ms/ALZ/CommunityCallAgenda

Microsoft





# This meeting is being recorded

# Before we get started...

At any point, if you have a question please put it in the chat!

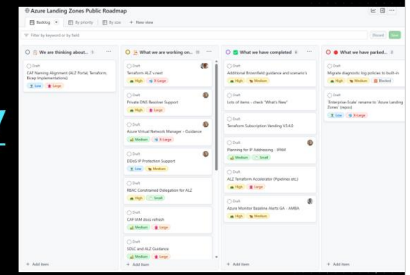
*(we have members of the team here to help 🥸)*

Also we may stop and discuss your question/point at that time, we want this to be an open discussion with all of you 🙂

Implementation Options & Accelerators

ALZ Public Roadmap

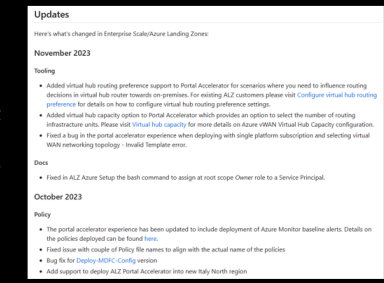
aka.ms/ALZ/Roadmap



ALZ What's New?

https://aka.ms/ALZ/WhatsNew

Single place to stay up-to-date




AMBA – It's GA!

MMA Deprecation Update

ALZ Terraform vNext Update

Asks to the community

Microsoft

Q & A

# Agenda

- Implementation Options & Accelerators Reminder
- Public Roadmap Reminder
- What's New in ALZ?
- AMBA Updates
- MMA Deprecation Updates
- ALZ Terraform v.next Intro & Deep Dive
- Our asks to you – help us drive the shape of ALZ
- Wrap up

# Implementation Options & Accelerators 🏗️ 🏎️

# Accelerators



- Azure Architecture Center
- Browse all Architectures
- Architecture icons
- What's new
- Landing zones
  - **Deployment Options**
  - Design guides
    - Landing zone implementations
      - Bicep landing zone implementation
      - Terraform landing zone implementation
    - Subscription vending implementation

## Cloud operating model roles and responsibilities

The Cloud Adoption Framework describes four common cloud operating models. The Azure identity and access for landing zones recommends five role definitions (Roles) you should consider if your organizations cloud operating model requires customized Role Based Access Control (RBAC). If your organization has more decentralized operations, the Azure built-in roles may be sufficient.

The table below outlines the key roles for each of the cloud operating models.

| Role | Decentralized operations | Centralized operations | Enterprise operations | Distributed operations |
|---|---|---|---|---|
| Azure platform owner (such as the built-in Owner role) | Workload team | Central cloud strategy | Enterprise architect in CCoE | Based on portfolio analysis - see Business alignment and Business commitments |
| Network management (NetOps) | Workload team | Central IT | Central Networking in CCoE | Central Networking for each distributed team + CCoE |
| Security operations (SecOps) | Workload team | Security operations center (SOC) | CCoE + SOC | Mixed - see: Define a security strategy |
| Subscription owner | Workload team | Central IT | Central IT + Application Owners | CCoE + Application Owners |
| Application owners (DevOps, AppOps) | Workload team | Workload team | Central IT + Application Owners | CCoE + Application Owners |

## Platform

The options below provide an opinionated approach to deploy and operate the Azure landing zone conceptual architecture as detailed in the Cloud Adoption Framework (CAF). It's important to note that, depending upon customizations, the resulting architecture might not be the same for all the options listed below. The differences between the options are how you deploy the architecture. They use differing technologies, take different approaches and are customized differently.

| Deployment option | Description |
|---|---|
| Azure landing zone Portal accelerator | An Azure portal-based deployment that provides a full implementation of the conceptual architecture, along with opinionated configurations for key components such as management groups and policies. |
| Azure landing zone Terraform accelerator | This accelerator provides an orchestrator module, but also allows you to deploy each capability individually or in part. |
| Azure landing zone Bicep accelerator | A modular accelerator where each module encapsulates a core capability of the Azure landing zone conceptual architecture. While the modules can be deployed individually, the design proposes the use of orchestrator modules to encapsulate the complexity of deploying different topologies with the modules. |

In addition, after deploying the landing zone, you will need to plan to operate it and maintain it. Review the guidance on how to Keep your Azure landing zone up to date.

## Subscription Vending

Once the platform landing zone is in place, the next step is to create and operationalize application landing zones for workload owners. Subscription democratization is a design principle of Azure landing zones that uses subscriptions as units of management and scale. This approach accelerates application migrations and new application development.

Subscription vending standardizes the process for requesting, deploying, and governing subscriptions, enabling application teams to deploy their workloads faster. To get started, see subscription vending implementation guidance, then review the following infrastructure-as-code modules. They provide flexibility to fit your implementation needs.

| Deployment option | Description |
|---|---|
| Bicep Subscription Vending | The Subscription Vending Bicep module is designed to accelerate deployment of the individual landing zones (aka Subscriptions) within an Azure Active Directory Tenant on EA, MCA & MPA billing accounts. |
| Terraform Subscription Vending | The Subscription Vending Terraform module is designed to accelerate deployment of the individual landing zones (aka Subscriptions) within an Azure Active Directory Tenant on EA, MCA & MPA billing accounts |

## Application

Application landing zones are one or more subscriptions that are deployed as environments for workloads or applications. These workloads can take advantage of services deployed in platform landing zones. The application landing zones can be centrally managed applications, decentralized workloads, or technology platforms such as Azure Kubernetes Service that host applications.

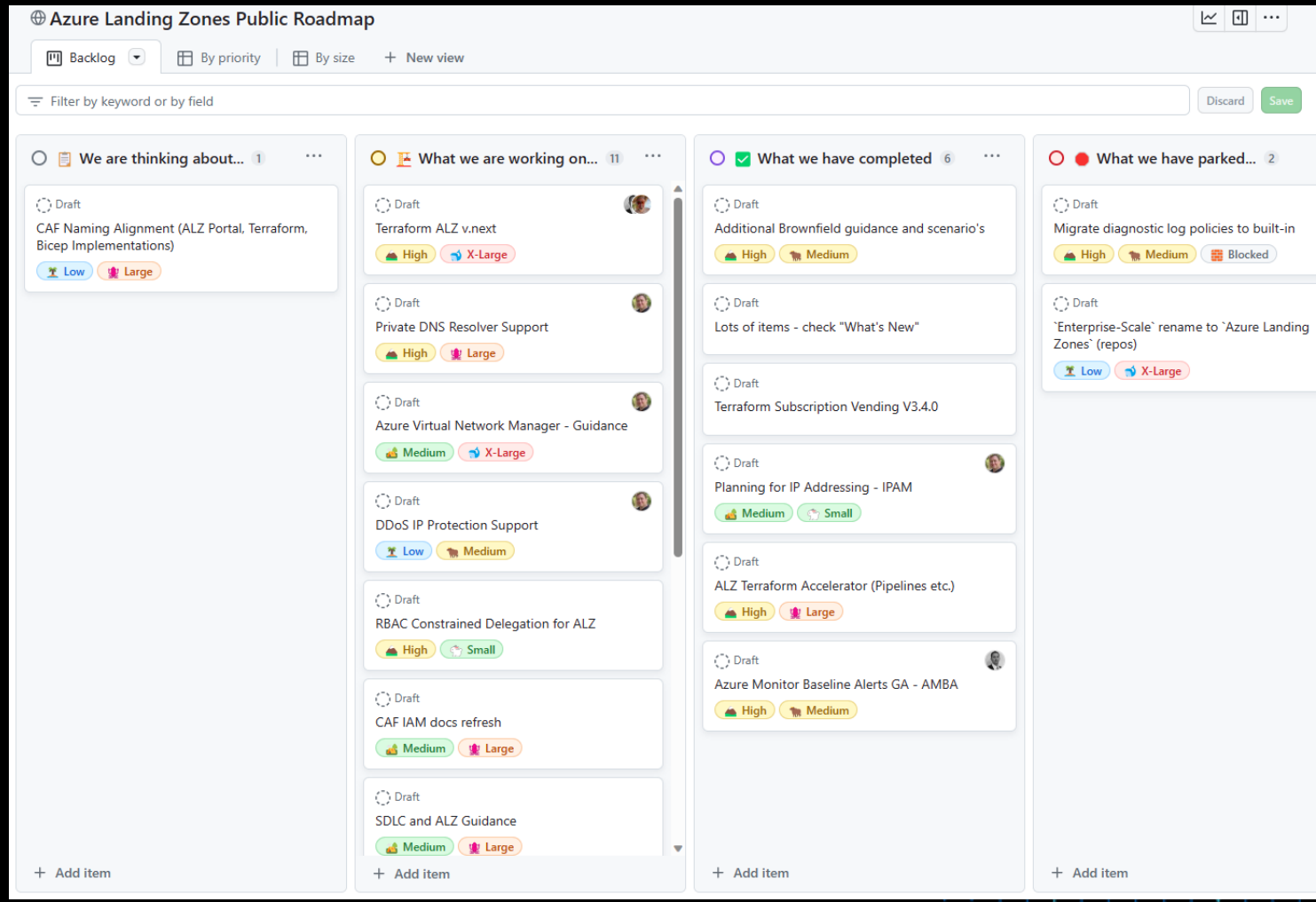
You can use the options below to deploy and manage applications or workloads in an application landing zone.

| Application | Description |
|---|---|
| AKS landing zone accelerator | An open-source collection of ARM, Bicep, and Terraform templates that represent the strategic design path and target technical state for an Azure Kubernetes Service (AKS) deployment. |
| Azure App Service landing zone accelerator | Proven recommendations and considerations across both multi-tenant and App Service Environment use cases with a reference implementation for ASEv3-based deployment |
| Azure API Management landing zone accelerator | Proven recommendations and considerations for deploying APIM management with a reference implementation showcasing App Gateway with internal APIM instance backed Azure Functions as backend. |
| SAP on Azure landing zone accelerator | Terraform and Ansible templates that accelerate SAP workload deployments using Azure Landing Zone best practices, including the creation of Infrastructure components like Compute, Networking, Storage, Monitoring & build of SAP systems. |
| HPC landing zone accelerator | An end-to-end HPC cluster solution in Azure using tools like Terraform, Ansible, and Packer. It addresses Azure Landing Zone best practices, including implementing identity, Jump-box access, and autoscale. |
| Azure VMware Solution landing zone accelerator | ARM, Bicep, and Terraform templates that accelerate VMware deployments, including AVS private cloud, jumpbox, networking, monitoring and add-ons. |
| Azure Virtual Desktop Landing Zone Accelerator | ARM, Bicep, and Terraform templates that accelerate Azure Virtual Desktop deployments, including creation of host pools, networking, storage, monitoring and add-ons. |
| Azure Red Hat OpenShift landing zone accelerator | An open source collection of Terraform templates that represent an optimal Azure Red Hat OpenShift (ARO) deployment that is comprised of both Azure and Red Hat resources. |
| Azure Arc landing zone accelerator for hybrid and multicloud | Arc enabled Servers, Kubernetes, and Arc-enabled SQL Managed Instance see the Jumpstart ArcBox overview. |

**aka.ms/ALZ/AAC**

# ALZ Public Roadmap

## aka.ms/ALZ/Roadmap




# ALZ What's New?

https://aka.ms/ALZ/WhatsNew

# Single place to stay up-to-date

## Updates

Here's what's changed in Enterprise Scale/Azure Landing Zones:

### November 2023

#### Tooling

- Added virtual hub routing preference support to Portal Accelerator for scenarios where you need to influence routing decisions in virtual hub router towards on-premises. For existing ALZ customers please visit Configure virtual hub routing preference for details on how to configure virtual hub routing preference settings.
- Added virtual hub capacity option to Portal Accelerator which provides an option to select the number of routing infrastracture units. Please visit Virtual hub capacity for more details on Azure vWAN Virtual Hub Capacity configuration.
- Fixed a bug in the portal accelerator experience when deploying with single platform subscription and selecting virtual WAN networking topology - Invalid Template error.

#### Docs

- Fixed in ALZ Azure Setup the bash command to assign at root scope *Owner* role to a Service Principal.

### October 2023

#### Policy

- The portal accelerator experience has been updated to include deployment of Azure Monitor baseline alerts. Details on the policies deployed can be found here.
- Fixed issue with couple of Policy file names to align with the actual name of the policies
- Bug fix for Deploy-MDFC-Config version
- Add support to deploy ALZ Portal Accelerator into new Italy North region

# Get in the (Landing) Zone with Terraform on Azure

ALZ Team @
HashiConf 2023


HashiCorp

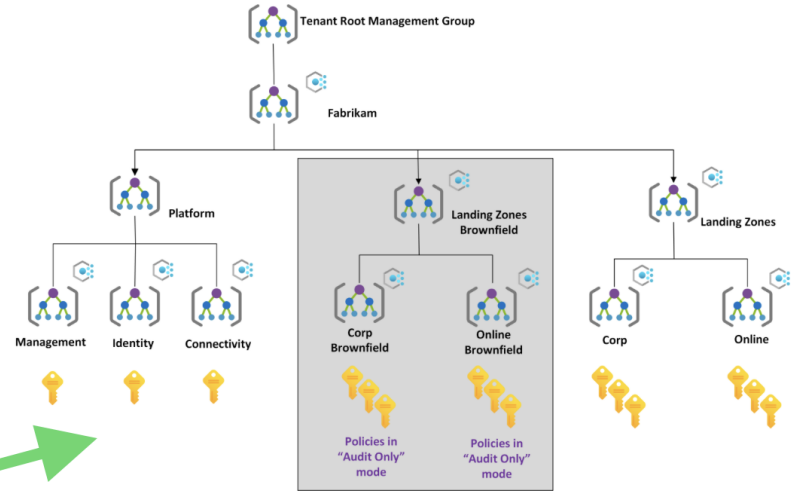
# More Brownfield Guidance

# NOW LIVE! aka.ms/alz/brownfield

Filter by title

- Align
  - Refactor landing zones
  - Transition to the Azure landing zone conceptual architecture
- Alignment scenarios
  - Single Subscription with no management groups to the Azure landing zone conceptual architecture
  - Single/Few Management Groups to the Azure landing zone conceptual architecture
  - Regional organization to the Azure landing zone conceptual architecture
- Alignment approaches
  - **Migration approach using "Audit Only" mode policies and hierarchy**
  - Brownfield landing zone considerations
- Enhance
- Azure landing zones FAQ

4. Duplicate the *Landing Zones* Management Group as well as it's children (Corp & Online), including all the policy assignments with configuring them to *audit only* mode, by setting the *Enforcement Mode* on the policy assignments to DoNotEnforce/Disabled. This approach allows getting into the new desired target architecture very quickly and then the applications teams can start to assess the policies applied without the risk of impacting any of the running applications.




5. (optional) Work with application or service teams to migrate the workloads deployed in the original subscriptions into new Azure subscriptions, per the guidance in Transition existing Azure environments to the Azure landing zone conceptual architecture. They can be placed into the newly duplicated management group hierarchy under the correct management group – *corp brownfield* or *online brownfield*.

# New Section in CAF

## Networking – AVNM

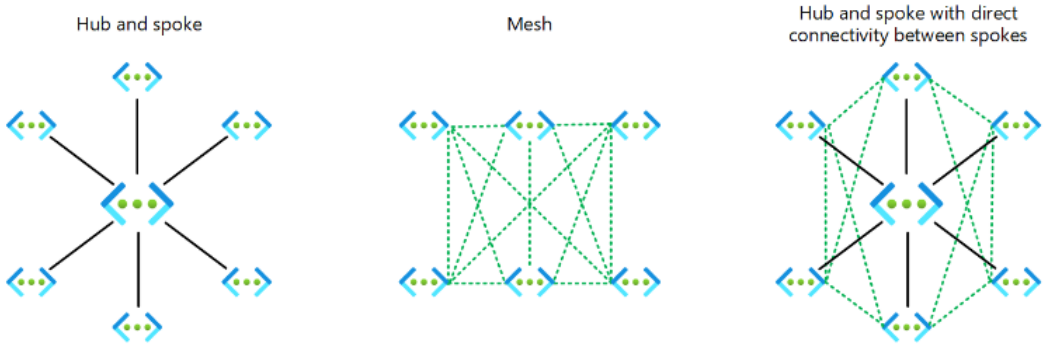
- Network topology and connectivity
  - Overview
  - Topology
    - **Define an Azure network topology**
    - Traditional Azure networking topology
    - Virtual WAN network topology (Microsoft-managed)
    - Plan for IP addressing

### Azure Virtual Network Manager in Azure Landing Zones

The Azure Landing Zones conceptual architecture recommends one of two networking topologies: an Azure Virtual WAN-based network topology or a network topology based on a traditional hub and spoke architecture. As the business requirements change over time (for example, migration of on-premises applications to Azure that requires hybrid connectivity), AVNM allows you to expand and implement networking changes, in many cases, without disrupting what is already deployed in Azure.

Azure Virtual Network Manager allows you to create three types of topologies across subscriptions for both existing and new virtual networks:

- Hub and spoke topology
- Hub and spoke topology with direct connectivity
- Mesh topology (Preview)




**Note**

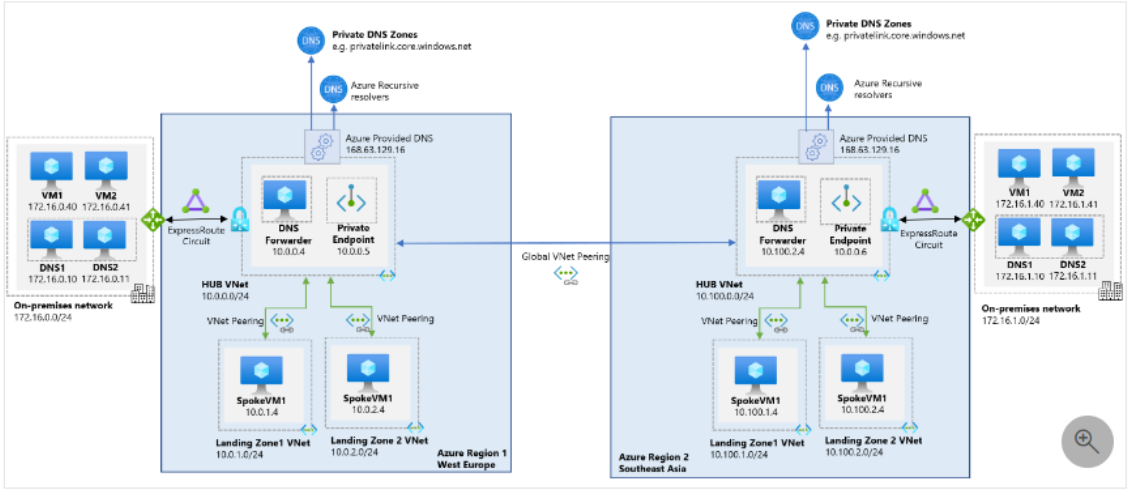
Azure Virtual Network Manager does not support Azure Virtual WAN hubs as part of a network group or as the hub in a topology. For more information, see **Azure Virtual Network Manager FAQ**.

# New Addition to CAF

# Networking – Multi-Region DNS

- Network topology and connectivity
  - Overview
  - Topology
  - Connectivity
  - Network security
  - Resources
    - **Private Link and DNS integration at scale**
    - DNS for on-premises and Azure resources
    - Plan for virtual machine remote access

While the previous diagram depicts a single hub and spoke architecture, customers might need to extend their Azure footprint across multiple regions to address resiliency, proximity or data residency requirements, several scenarios have emerged where the same Private-Link-enabled PaaS instance must be accessed through multiple Private Endpoints (PE's).




The following diagram shows a typical high-level architecture for enterprise environments with central DNS resolution deployed in the hub (one per region) where name resolution for Private Link resources is done via Azure Private DNS.

It is recommended to deploy multiple regional private endpoints associated to the PaaS instance, one in each region where clients exist, enable per-region Private Link and Private DNS Zones. When working with PaaS services with built-in DR capabilities (geo-redundant storage accounts, SQL DB failover groups, etc.), multiple region Private Endpoints are mandatory.

This scenario requires manual maintenance/updates of the Private Link DNS record set in every region as there is currently no automated lifecycle management for these.

For other use cases, a single global Private Endpoint can be deployed, making accessible to all clients by adding routing from the relevant regions to the single Private Endpoint in a single region.

# COMING SOON

# Application development environments in Azure landing zones

## Application development environments in Azure landing zones

Article • 12/01/2023 • 3 contributors

Feedback

**In this article**

Environments, Subscriptions, and Management Groups
Next steps

Development teams want limited interference with the ability to iterate quickly, while cloud governance and platform teams need to solve for organizational risk, compliance, and security at scale. Azure landing zone's design principles guide customers to adopt policy-driven governance and Subscription democratization as two key principles. These provide foundational guard rails, while also delegating many controls to application teams. These applications teams design their workload using guidance from the Azure Well-Architected Framework, then deploy and manage their own landing zone resources; whilst being controlled by Azure policies assigned by the platform team.

A key part of this approach is to provide sandbox resources for "semi-governed" resources that allow for application teams to explore technologies and capabilities.

This article provides guidance for how cloud platform teams can provide guardr
in Azure. The article gives guidance on how an organization can fit different app
this framework. A key aspect is placing environment subscriptions in appropriat

When application owners use Subscription Vending or other subscription creatic
be clarity about how to request subscriptions for multiple development environ

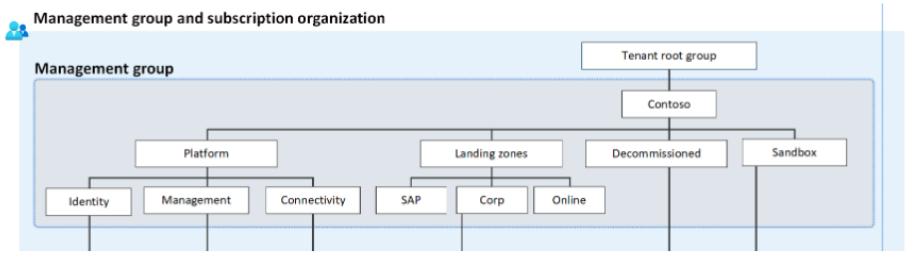
In this guide, we talk both about the Azure landing zone - the management gro
architecture - and the workload or application landing zone.

ⓘ Note

This is for workload landing zones only. For testing and environment segrega
platform itself, review the **testing approach for enterprise-scale** (Canary appr

## Challenges with having environment based management groups

Building management groups for for environments within the archetypes creates more management overhead, while providing little value.




The *Landing Zone* management group should have universal polices that enforce guardrails for both Corp and Online. Corp and Online have unique polices to enforce company guidelines around public and private facing workloads.

Many organizations think that they should create management groups for different workload SDLC environments and assign environmental policies and controls. In practice, this creates more challenges for workload teams than it solves. Policies should not differ between SDLC environments, and so separate management groups for those environments is not suggested.

For application owners, there's significant risk in changing the topology or resource configuration of a workload as it's promoted through the different SDLC environments just to align to policies specific to that environment. Per-environment rules results in a poor development experience for developer and quality assurance teams. If an application is built with one set of guardrail policies that works in one environment but is exposed to a different set later in its promotion cycle, this can create issues. Applications might have to be reworked due to changing controls.

To prevent this rework, policies should remain consistent throughout promotion of code through SDLC environments. Platform teams should not build policies for each environment, but instead provide a consistent set for all non-Sandbox development environments.

# ALZ Policy Testing Framework

## Wiki content

- What's New?
- Community Calls
- Frequently Asked Questions (FAQ)
- Known issues
- What is Enterprise-Scale
  - Architecture
  - Policies
  - Policies FAQ & Tips
  - Policies Testing Framework
  - What is the reference implementation?
  - Pricing
  - What if I already have an existing Azure footprint

## Azure Landing Zone Policy Testing Framework

### Overview

The ALZ Policy Testing Framework is a set of tools and scripts that can be used to test Azure Policies do what is expected and prevent breaking regressions. The framework is designed to be used with pipelines as part of CI/CD processes to test policies as they are developed and integrated to ultimately improve the quality and stability of policies going into production environments.

This framework is based on the work done by @fawohlsc in this repo azure-policy-testing, and is built on the well established PowerShell testing frame

For ALZ, the focus is on t
organizations if a regress
production environments
framework.

> *NOTE:* The ALZ team a
> DeployIfNotExists.

For authoring tests we sta
implement and read, how
through Az PowerShell. T
calls and make it easier t

### Prerequisites

- An empty (dedicated
  - If following the s
    management gr
- Pester
- Az PowerShell Modu
- Invoke-AzRestMetho

### How to write Pester tests for policies

For the purposes of this guide, we'll focus on the Policy test for `Deny-MgmtPorts-Internet` policy as it demonstrates using both Az PowerShell and REST API calls in the Pester test. The policy definition file is located in the `policy` folder of the ALZ repository in the `policy` folder.

The policy tests are designed to run in an empty subscription(s) to ensure that the policy is evaluated in isolation and not impacted by other policies or resources in the subscription.

> *NOTE:* Because we are testing Azure policies in the context of Azure Landing Zone, we are using a dedicated subscription in the "Corp" landing zone that is added under the Corp management group, where we retrieve the deployed policy definition ID and create a new policy assignment to test the policy (because we do not assign all policies by default, and some get assigned to different scopes). You can extend this methodology to test policies outside of Azure Landing Zone by deploying the policy etc.

The policy test has 4 main

**BeforeAll: This section is u**

```
# Set the defau
Set-AzContext -

if (-not [Strin
    Write-Infor
    $deployment

    # Set the e
    $esCompanyP
    $mangementG
}

$definition = G
New-AzPolicyAss
    "ports" = @
}
```

If you want to extend this methodology to test policies independent of deploying ALZ, you could extend this section to also deploy the policy you want to test, and then do the policy assignment.

**DENY - group of tests to validate scenarios that where the policy effect is applied and deployment should fail.**

As an example, using Az PowerShell:

```
It "Should deny non-compliant port '3389'" -Tag "deny-noncompliant-nsg-port" {
    AzTest -ResourceGroup {
        param($ResourceGroup)

        $networkSecurityGroup = New-AzNetworkSecurityGroup `
        -Name "nsg-test" `
        -ResourceGroupName $ResourceGroup.ResourceGroupName `
        -Location $ResourceGroup.Location

        # Should be disallowed by policy, so exception should be thrown.
        {
            $networkSecurityGroup | Add-AzNetworkSecurityRuleConfig `
                -Name RDP-rule `
                -Description "Allow RDP" `
                -Access Allow `
                -Protocol Tcp `
                -Direction Inbound `
                -Priority 200 `
                -SourceAddressPrefix * `
                -SourcePortRange * `
                -DestinationAddressPrefix * `
                -DestinationPortRange 3389 # Incompliant.
            | Set-AzNetworkSecurityGroup
        } | Should -Throw "*disallowed by policy*"
    }
}
```

In this example, we are creating a new Network Security Group (NSG) and adding a rule to allow RDP traffic on port 3389. The policy we're testing is configured to deny traffic on port 3389, so we expect this operation to fail. We use the `Should -Throw` command to validate that the operation failed with the expected error message.

# ALZ Portal Updates

aka.ms/alz/portal

- Virtual WAN Routing Intent Support – Thanks Recep 👏
- VPN GW Active/Active Config Support – Thanks Terry 👏
- Azure Firewall Basic IP Subnet Clashes & Bug When Deployed With VPN GWs – Thanks Sacha & Jack 👏
- Support Added For Italy North Region – Thanks Jack & Jared 👏
- AMBA Added As New Blade – Thanks Jan & Arjen 👏

# ALZ Policy News

aka.ms/alz/policies

- New Backup initiative and updates
  - Immutability audit
  - Storage account backup
  - Vault private endpoints
  - Working with PG to create new policies
  - *New Assignment*
- New Zone Resilience initiative
  - Audit Zone Resiliency based for supporting resources
  - *New Assignment*
- Diagnostics Settings
  - Dependency on versioning (delayed)
  - Looking for a way forward in short term

# AMBA – It’s GA!

# AMBA Updates

aka.ms/amba/patterns/alz

- Finished migration to the new AMBA repo. It´s GA! - Thanks Jan, Sacha, Bruno, Alex, Paul, Bryan, Joseph, Arjen 👏
- Added to ALZ Portal deployment – Thanks Jan & Sacha 👏
- Service Health Granular Deployment Flexibility – Thanks Bruno 👏
- Documentation updates – Thanks Bryan 👏
- Added guidance on how to deploy only Service Health alerts - Thanks Arjen 👏
- Added support for adding multiple email address in Action Group – Thanks Bruno 👏
- New alerts – Thanks Alex 👏
  - Application gateway
  - Load balancers
  - Express route ports

# AMBA Roadmap

aka.ms/amba/patterns/alz

- Landing zone initiative v2
  - Front door (classic)
  - Front door (CDN profiles)
  - App Services
  - Traffic Manager
- Decoupling Service Health initiative deployment from Action Group and Alert Processing rules
- Extending existenceCondition to detect and remediate configuration drift
- ALZ Portal updates
- Action Group Actions and Notifications flexibility
  - Web and secure webhook
  - Logic Apps
  - Event hub
  - ARM roles
  - Azure Function
  - Automation Runbook
- AMBA for ALZ Bicep/ Terraform

# MMA Deprecation Update

# MMA → AMA or Alternatives Update

- We have been working on this for a while – over a year 🤯
  - And it's still a work in progress – it's complex

- UMC, Change Tracking, Sentinel, MDFC (and all its plans), VM Insights, Policies, Managed Identities etc.
  - We've had policies fixed that had hardcoded regions to East US, hardcoded VM SKU lists and more

- We think we might finally have all the pieces of the puzzle to piece it together for ALZ customers
  - MDFC is the last piece of the puzzle and we're working closely with the PG
  - April 2024 for MDFC for VMs parity to MMA (best case timeline, may slip)

# ALZ Terraform vNext Update

# ALZ Terraform Module vNext

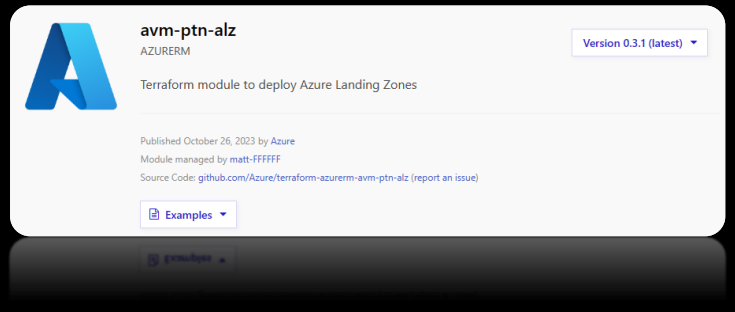
New Pattern Module: aka.ms/avm-ptn-alz

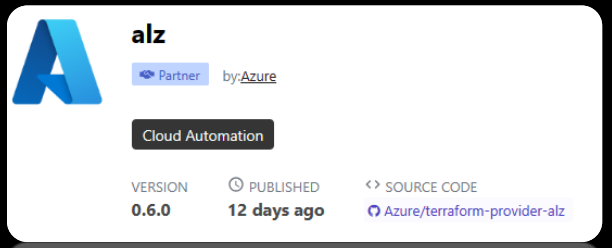
(Backed by the ALZ Terraform Provider: aka.ms/tf-pdr-alz)

Example usage: aka.ms/avm-ptn-alz-example

## Scope of the new module

- Management groups
- Policy
- Role Assignments



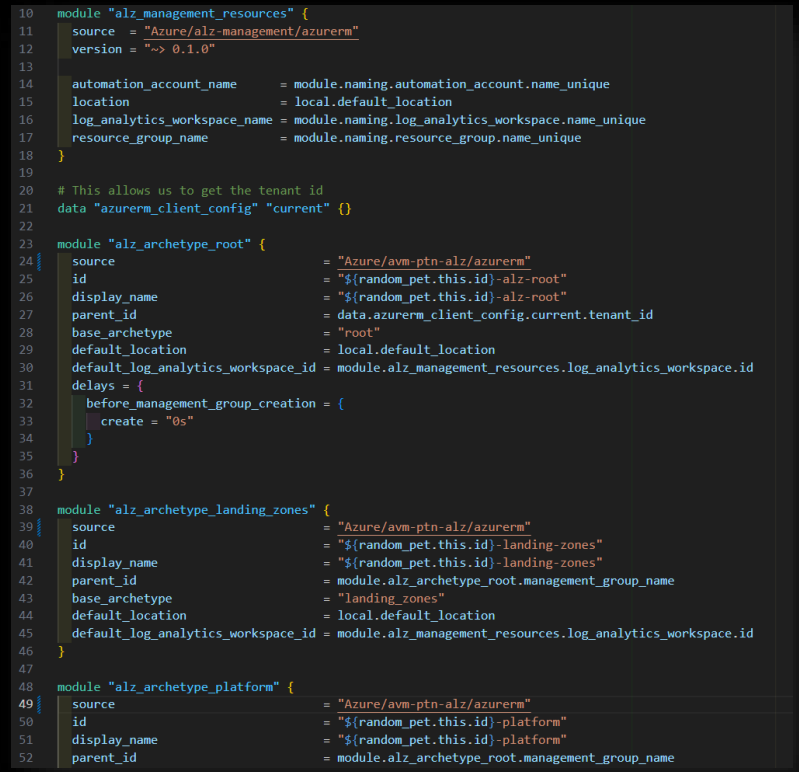
# How to build a Landing Zone

## Compose the modules together

- Azure/avm-ptn-alz: Management groups and policy
- Azure/alz-management: Management resources like log analytics, etc
- Azure/hubnetworking: Hub networking with vNet or vWan
- Azure/vnet-gateway: vNet gateways
- Etc…

## Azure Verified Modules (AVM)?

- These modules will be renamed to align with AVM soon!



```
module "alz_management_resources" {
  source  = "Azure/alz-management/azurerm"
  version = "~> 0.1.0"

  automation_account_name      = module.naming.automation_account.name_unique
  location                     = local.default_location
  log_analytics_workspace_name = module.naming.log_analytics_workspace.name_unique
  resource_group_name          = module.naming.resource_group.name_unique
}

# This allows us to get the tenant id
data "azurerm_client_config" "current" {}

module "alz_archetype_root" {
  source                            = "Azure/avm-ptn-alz/azurerm"
  id                                = "${random_pet.this.id}-alz-root"
  display_name                      = "${random_pet.this.id}-alz-root"
  parent_id                         = data.azurerm_client_config.current.tenant_id
  base_archetype                    = "root"
  default_location                  = local.default_location
  default_log_analytics_workspace_id = module.alz_management_resources.log_analytics_workspace.id
  delays = {
    before_management_group_creation = {
      create = "0s"
    }
  }
}

module "alz_archetype_landing_zones" {
  source                            = "Azure/avm-ptn-alz/azurerm"
  id                                = "${random_pet.this.id}-landing-zones"
  display_name                      = "${random_pet.this.id}-landing-zones"
  parent_id                         = module.alz_archetype_root.management_group_name
  base_archetype                    = "landing_zones"
  default_location                  = local.default_location
  default_log_analytics_workspace_id = module.alz_management_resources.log_analytics_workspace.id
}

module "alz_archetype_platform" {
  source                            = "Azure/avm-ptn-alz/azurerm"
  id                                = "${random_pet.this.id}-platform"
  display_name                      = "${random_pet.this.id}-platform"
  parent_id                         = module.alz_archetype_root.management_group_name
```

# Demo and Roadmap


## Can I use it now?

- Yes. We want you to try it out now!
- Give us feedback by raising issues and / or PRs in GitHub.

## What's coming next?

- Improvements based on your and internal feedback
- Subscription placement
- Migration path
- Documentation
- End to end example in the ALZ Terraform Accelerator: [aka.ms/alz-tf-acc](aka.ms/alz-tf-acc)

## What does it look like?

- Demo Time!

# Asks to the community

# Help us shape the ALZ roadmap with these two short surveys!

1. **Enabling multi-region support out of the box for ALZ**
   - Complete/fill out this survey for us

2. **Tell us about how you deploy workloads!**
   - Complete/fill out this survey for us (takes two minutes)

Microsoft

# Q & A

Next Community Call will be in 11th March 2024 👍

Back to an US friendly time slot for this occurrence and then the one after will be back to this time slot 👍

Stay tuned to issue #1491 (ALZ/ESLZ Repo)

Recordings will be available at:
aka.ms/ALZ/Community

This month's presenters:

# Thank You! 👋

**Stay up-to-date:**
**https://aka.ms/ALZ/WhatsNew**