### **What is BigID?**

BigID is a data intelligence platform designed for data discovery, classification, cataloging, and management. It enables organizations to automatically identify, classify, and manage sensitive, personal, and regulated data across various data sources, including cloud environments like Azure Blob Storage. BigID uses advanced techniques such as pattern matching, machine learning (ML)-based classification, and metadata analysis to provide comprehensive visibility into data assets.

### **Benefits of BigID**

* **Compliance and Risk Mitigation**: Supports regulatory compliance (e.g., GDPR, CCPA) by discovering and classifying personal and sensitive data, helping to fulfill data rights requests and reduce breach risks.
* **Data Governance**: Minimizes duplicate and redundant data, integrates with third-party catalogs (e.g., Alation, Collibra), and provides a unified view of data across environments.
* **Security and Privacy**: Automates scanning of data at rest or in motion, enabling proactive privacy preservation and security posture improvements.
* **Efficiency in Cloud Operations**: Simplifies metadata management, eases cloud migrations, and offers flexible deployment options like Azure Marketplace integration.
* **Scalability**: Handles large-scale environments, including multi-subscription and multi-region setups in Azure, with customizable scanning modes (e.g., full, differential, or hyperscans).

Implementing BigID in Azure to scan blob storage accounts across multiple subscriptions and regions (East US, West US 2, Southeast Asia, Central India) involves deploying BigID via the Azure Marketplace, using a service principal for authentication, creating a custom IAM role for least-privilege access, and configuring storage account networking firewalls to allow access from the BigID VM's subnet. The best approach uses BigID's Azure connector for scanning, ensuring centralized management while respecting regional data residency. Below is a step-by-step guide based on best practices.

### **Step-by-Step Implementation**

#### **1. Plan Your Architecture**

* **Subscriptions**: List all Azure subscriptions with blob storage accounts to scan. BigID can scan across subscriptions using a single service principal assigned roles at the management group or subscription level.
* **Regions**: Deploy BigID in a central region (e.g., East US) for management, but ensure it can access storage accounts in East US, West US 2, Southeast Asia, and Central India via Azure's global backbone. Storage accounts are region-specific, so scans are API-driven and not limited by BigID's deployment region.
* **BigID Deployment**: Use Azure Marketplace to deploy BigID as a VM (recommended for control) or via Docker/Kubernetes for scalability.
* **Azure Requirements**:
  + **Service Principal**: For non-interactive authentication to Azure APIs.
  + **Custom IAM Role**: A least-privilege role with read permissions for blob storage.
  + **Storage Account Networking**: Configure firewalls to allow traffic from the BigID VM's virtual network (VNet) subnet.
* **Permissions Needed**: BigID requires read access to storage accounts (e.g., Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read).

#### **2. Set Up Azure Components**

##### **a. Create a Service Principal**

* **Purpose**: Authenticates BigID to Azure without user credentials.
* **Steps**:
  + In the Azure Portal, go to **Microsoft Entra ID > App Registrations > New Registration**.
  + Name it (e.g., BigIDScanner), select "Accounts in this organizational directory only," and register.
  + Note the **Application (Client) ID** and **Directory (Tenant) ID**.
  + Under **Certificates & Secrets**, create a new client secret (set to expire in 1-2 years) and copy the value.
  + (Optional) Upload a certificate for more secure authentication.

##### **b. Create a Custom IAM Role**

* **Purpose**: Provides least-privilege access for scanning blob storage data.
* **Steps**:
  + In the Azure Portal, go to **Subscriptions > [Your Subscription] > Access Control (IAM) > Roles > Add > Add Custom Role**.
  + Name it (e.g., BigIDBlobReader), add a description (e.g., "Read-only access for BigID scanning of blob storage").
  + Start from a built-in role like "Storage Blob Data Reader" and clone it.
  + Under **Permissions**, ensure it includes:
    - Microsoft.Storage/storageAccounts/listKeys/action (to access keys if needed).
    - Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read.
    - Microsoft.Storage/storageAccounts/read (for listing accounts).
    - Add data actions like Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read for blob content access.
  + Remove unnecessary permissions to follow least-privilege.
  + Set **Assignable Scopes** to include all target subscriptions (e.g., /subscriptions/{subId1}, /subscriptions/{subId2}) or a management group for multi-subscription assignment.
  + Review and create the role.
  + Assign the role to the service principal:
    - Go to **Access Control (IAM) > Add > Add Role Assignment**.
    - Select the custom role, assign to the service principal, and apply at subscription or management group scope.

##### **c. Configure Storage Account Networking Firewall**

* **Purpose**: Restrict access to blob storage, allowing only the BigID VM's subnet.
* **Steps** (for each storage account in all subscriptions and regions):
  + In the Azure Portal, go to the storage account > **Networking > Firewalls and Virtual Networks**.
  + Set **Public Network Access** to "Enabled from selected virtual networks and IP addresses."
  + Under **Virtual Networks**, click **+ Add Existing Virtual Network** or **+ Add New Virtual Network**.
    - Select the VNet and subnet where the BigID VM is deployed (e.g., if BigID VM is in East US VNet/subnet, add that).
    - Enable the **Microsoft.Storage** service endpoint on the subnet for secure connectivity (or **Microsoft.Storage.Global** for cross-region access).
  + (Optional) Add the BigID VM's public IP if using hybrid access, but prefer VNet rules for security.
  + Save changes. This allows traffic only from the specified subnet, denying all others.
  + Repeat for storage accounts in East US, West US 2, Southeast Asia, and Central India. Use Azure CLI for automation:az storage account network-rule add \  
     --resource-group <rg> \  
     --account-name <storage-account> \  
     --vnet-name <vnet> \  
     --subnet <subnet>
  + Verify by testing access from the BigID VM subnet.

#### **3. Deploy and Configure BigID**

* **Deployment**:
  + Search for "BigID" in the Azure Marketplace.
  + Select the BigID offering (e.g., VM image) and deploy to a VM in your chosen region (e.g., East US for centrality).
  + Configure VM size based on scan volume (e.g., Standard\_D4s\_v5 for moderate loads), attach to the planned VNet/subnet, and set up networking.
* **Configuration for Scanning**:
  + Log in to the BigID console (web interface post-deployment).
  + Go to **Data Sources > Add Data Source > Cloud > Azure**.
  + Enter the service principal details: Tenant ID, Client ID, Client Secret.
  + Select scopes: All subscriptions or specific ones.
  + Choose data types: Focus on Blob Storage; enable classification for sensitive data (e.g., PII).
  + Configure scan settings: Full scan for initial baseline, then differential for updates. Set regions explicitly if needed (BigID handles multi-region via API).
  + Test connection to verify access to blob storage across regions.
  + Schedule scans (e.g., daily) and set up alerts for discoveries.

#### **4. Handle Multi-Region and Multi-Subscription Challenges**

* **Multi-Subscription**: Assign the custom role to the service principal at a management group level to cover all subscriptions. In BigID, add multiple Azure accounts if needed.
* **Multi-Region**: BigID scans via Azure APIs, so no additional regional deployments are required. Ensure storage account firewalls allow the central BigID subnet (use cross-region service endpoints). For data residency, deploy regional BigID instances if compliance demands it.
* **Automation**: Use ARM templates to deploy consistent configurations:{  
   "type": "Microsoft.Authorization/roleDefinitions",  
   "properties": {  
   "roleName": "BigIDBlobReader",  
   "permissions": [  
   {  
   "actions": ["Microsoft.Storage/\*/read"]  
   }  
   ],  
   "assignableScopes": ["/subscriptions/{subId1}", "/subscriptions/{subId2}"]  
   }  
  }
* **Scalability**: For high volumes, use BigID's hyperscan mode and monitor Azure API throttling.

#### **5. Security and Networking**

* **Service Principal Security**: Use certificate auth over secrets; rotate credentials regularly.
* **Firewall Best Practices**: Enable "Allow trusted Microsoft services" if BigID integrates with other Azure tools. Log denied access for auditing.
* **Encryption**: Ensure blob storage uses Azure-managed keys; BigID supports scanning encrypted data.
* **VNet Peering**: If BigID VM is in one region, peer VNets for cross-region access if needed.

#### **6. Verify and Troubleshoot**

* **Test Scans**: Run a sample scan in BigID; check for classified data in the dashboard.
* **Logs**: Review BigID logs for errors (e.g., access denied) and Azure Activity Logs for role assignments.
* **Troubleshooting**:
  + Access Issues: Verify role assignments and firewall rules.
  + Multi-Region Delays: Check API limits; use Azure Monitor.
  + No Data: Ensure service principal has correct scopes.
* **Costs**: Monitor VM compute, storage egress, and BigID licensing.

### **Best Practices**

* **Least Privilege**: Limit the custom role to read-only for scanning.
* **Monitoring**: Integrate BigID with Azure Monitor for alerts.
* **Backup**: Configure BigID for high availability across regions.
* **Updates**: Regularly update BigID via Marketplace for new features.

### **Limitations**

* **Regional Access**: API-based scans may incur latency in distant regions like Southeast Asia from East US.
* **Data Volume**: Large blobs may require optimized scan modes to avoid timeouts.

For more on BigID, visit [https://bigid.com](https://bigid.com/). If you need a diagram of the architecture or xAI API for automation, let me know!