

Advanced EC2

Bootstrapping EC2 using User Data

- Bootstrapping
 - Brings automation into EC2.
 - The process of running scripts or config when an instance first launches.
 - Automates software installs and applies post-launch configuration.
- Bootstrapping is enabled via EC2 User Data.
- User data is a block of data passed to an instance at launch.
- It's executed only once—on first launch only.
 - If the instance is restarted or user data is changed, it won't re-run automatically.
 - Delivered via metadata IP: 169.254.169.254/latest/user-data.
 - EC2 doesn't validate or interpret it.
 - Just runs it as root.
- Bootstrapping Flow
 - Launch EC2 instance using an AMI
 - Attach EBS volume based on block device mapping
 - EC2 passes user data to the instance
 - The OS checks for user data at the metadata endpoint
 - If present, the user data is executed as a script on first boot
 - If user data fails, the instance still boots, but may be misconfigured.
- Security and Risk Considerations
 - No validation
 - EC2 blindly passes data
 - Runs as root
 - Can cause issues if misused (e.g., delete boot volume)
 - Not secure
 - Anyone with OS access can read user data
- 16 KB size limit
- Boot Time vs. Service Time
 - Boot Time: Time AWS takes to provision the EC2 instance
 - Post-Launch Time: Time config or app installs take to complete
 - Service Time: When the instance is fully ready for traffic; boot time + post-launch time

- Example: If an instance is ready in 3 minutes, it includes:
 - AWS booting the instance
 - Any user data scripts running (e.g., installing/configuring software)
- Ways to Reduce Post-Launch Time
 - Bootstrapping
 - No human delay
 - Scalable
 - AMI Baking
 - Pre-configure an AMI with installed applications and system packages and settings
 - Both methods can be used together:
 - Baking for time-consuming installations steps
 - Bootstrap for quick configurations

Bootstrapping with CFN-INIT

- CloudFormation offers declarative, state-driven configuration.
 - “State-driven” because CloudFormation doesn’t just run commands.
 - It evaluates the desired infrastructure.
 - It ensures that the desired state is met.
- Bootstrapping with cfn-init
 - cfn-init: A helper script that runs on EC2 instances launched via AWS CloudFormation.
 - Pulls setup instructions from the CloudFormation stack metadata.
 - cfn-init is used to process metadata defined in the CloudFormation template under the `AWS::CloudFormation::Init` section.
 - Applies configuration to the EC2 instance.
 - Unlike User Data, cfn-init ensures that the desired state is met.
- Execution Flow
 - Launch a CloudFormation stack with an EC2 instance.
 - The EC2 instance boots with a User Data script that calls cfn-init.
 - cfn-init fetches metadata from the stack.
 - Configuration is applied (e.g., installing software such as Apache).
 - cfn-signal is sent back to CloudFormation to confirm success or failure.
 - cfn-signal: Reports success/failure of the bootstrapping process.

- Creation Policies: Instruct CloudFormation to wait for a signal before marking the resource as complete.

EC2 Instance Roles & Profile

- EC2 instance roles are a type of IAM role specifically for EC2 instances.
- Any application running on the EC2 instance inherits the role's permissions automatically.
- Architecture
 - IAM Role: Contains a permission policy.
 - When assumed, it provides temporary credentials based on the permission policy.
 - Instance Profile: A wrapper around the IAM role.
 - It's what is actually attached to an EC2 instance.
 - AWS console automatically creates the instance profile and has the same name as the role.
 - But, in CLI or CloudFormation, you have to create the IAM role and instance profile separately.
- Credentials are delivered via metadata.
 - Metadata contains temporary credentials.
 - Credentials are used to access AWS services.
 - Credentials are auto-rotated before expiry by EC2 and Secure Token Service.

Systems Manager (SSM) Parameter Store

- SSM securely store configuration data such as documents and passwords.
- A bad practice is to embed secrets in User Data.
- Parameter Types
 - String: Plain texts
 - StringList
 - SecureString: Encrypted with KMS, for storing sensitive data like passwords.
- Each change to a parameter creates a new version.
- Enables rollback and tracking.
- Integrated with IAM.

System and Application Logging on EC2

- CloudWatch is used for storing and managing metrics.
- CloudWatch Logs is designed for storing, managing, and visualizing logs.
- By default, CloudWatch cannot access OS-level data inside EC2 instances.
- CloudWatch Agent is installed inside the instance to gain visibility inside an EC2 instance.
 - Collects performance metrics and system/application logs.
 - Sends collected data to CloudWatch and/or CloudWatch Logs.
 - IAM role can be attached to grant an agent access to an EC2 instance.
- Can be automated using CloudFormation, User Data, and SSM.

EC2 Placement Groups

- The physical placement of EC2 instances within an AZ to optimize performance or resilience.
- 3 Types of Placement Groups

Type	Goal	Key Feature
Cluster	Performance	Instances placed close together
Spread	High availability	Instances placed far apart on different racks
Partition	Resilient, large-scale apps	Instances grouped in isolated partitions

- Cluster Placement Group
 - Maximum performance (high throughput and low latency)
 - Instances placed on the same rack or EC2 host
 - High single stream bandwidth up to 10 Gbps
 - Requires enhanced networking enabled on instances
 - Cluster group is locked to a single AZ
 - Not supported by all instance types
 - Low fault tolerance
 - Use cases
 - High Performance Computing (HPC)
 - Distributed computing (requires fast node-to-node communication)

- Spread Placement Group
 - Maximum resilience and availability
 - Instances placed on separate racks
 - Supports multiple AZs
 - Cannot use Dedicated Instances or Dedicated Hosts
 - Low performance
 - Small scale (7 instances/AZ)
- Partition Placement Group
 - Supports large-scale, topology-aware applications for fault tolerance
 - Isolated groups of instances
 - Topology-aware application
 - Aware of the physical or logical structure of the infrastructure that it runs on
 - Each partition uses isolated racks
 - Up to 7 partitions/AZ
 - More complex than Spread
 - Use case
 - Large distributed systems with internal replication
 - Better control over failure
 - Enables applications to replicate data intelligently across isolated groups.

Dedicated Hosts

- A physical server fully dedicated to your AWS account.
 - Renting the entire host
 - Pay for host, not per instance
- Payment Options
 - On-Demand: Flexible, short-term use
 - Reserved: 1- or 3-year commitment; pay upfront, partial upfront, or no upfront
- Useful for software licensed per socket or core (e.g., Oracle, SQL Server)
- A host can be shared with other AWS accounts in the same Organization.
 - Can only view/control the instance the account creates
 - The host owner can see all instances running on the host
 - But, no control on the instances created by other AWS accounts

Enhanced Networking & EBS Optimized

- Features that improve EC2 instance networking and storage performance.
- Enhanced Networking
 - Improves EC2 network throughput, latency, and packets per second (PPS) by reducing overhead and offloading work from the EC2 host.
 - Uses SR-IOV (Single Root I/O Virtualization)
 - Creates logical network interfaces that are directly mapped to EC2 instances.
 - The physical NIC (Network Interface Card) is virtualization-aware and offloads most of the processing.
 - Without Enhanced Networking:
 - EC2 host mediates all networking.
 - Shared access to a single physical NIC.
 - Networking handled by host software
 - Slower
 - Higher CPU overhead
 - Potential latency spikes
 - With Enhanced Networking:
 - Instances get dedicated logical interfaces.
 - Offloads network traffic directly to the NIC.
 - Reduces CPU usage on the host.
 - Better bandwidth, PPS, and latency (and consistency of latency)
 - Required for Cluster Placement Groups.
 - Supported on most modern instance types, usually enabled by default and free.
- EBS-Optimized Instances
 - Elastic Block Storage provides network-attached block storage for EC2.
 - EC2 networking was shared between:
 - General network traffic
 - EBS storage traffic
 - The sharing caused network contention, reducing performance.
 - EC2 instance gets dedicated bandwidth for EBS traffic.
 - Results in faster and more consistent EBS performance.
 - Higher throughput and IOPS.

- Supported on most modern instance types, enabled by default with no extra cost.