AWS Foundation

Introduction to IAM & CloudWatch

Agenda

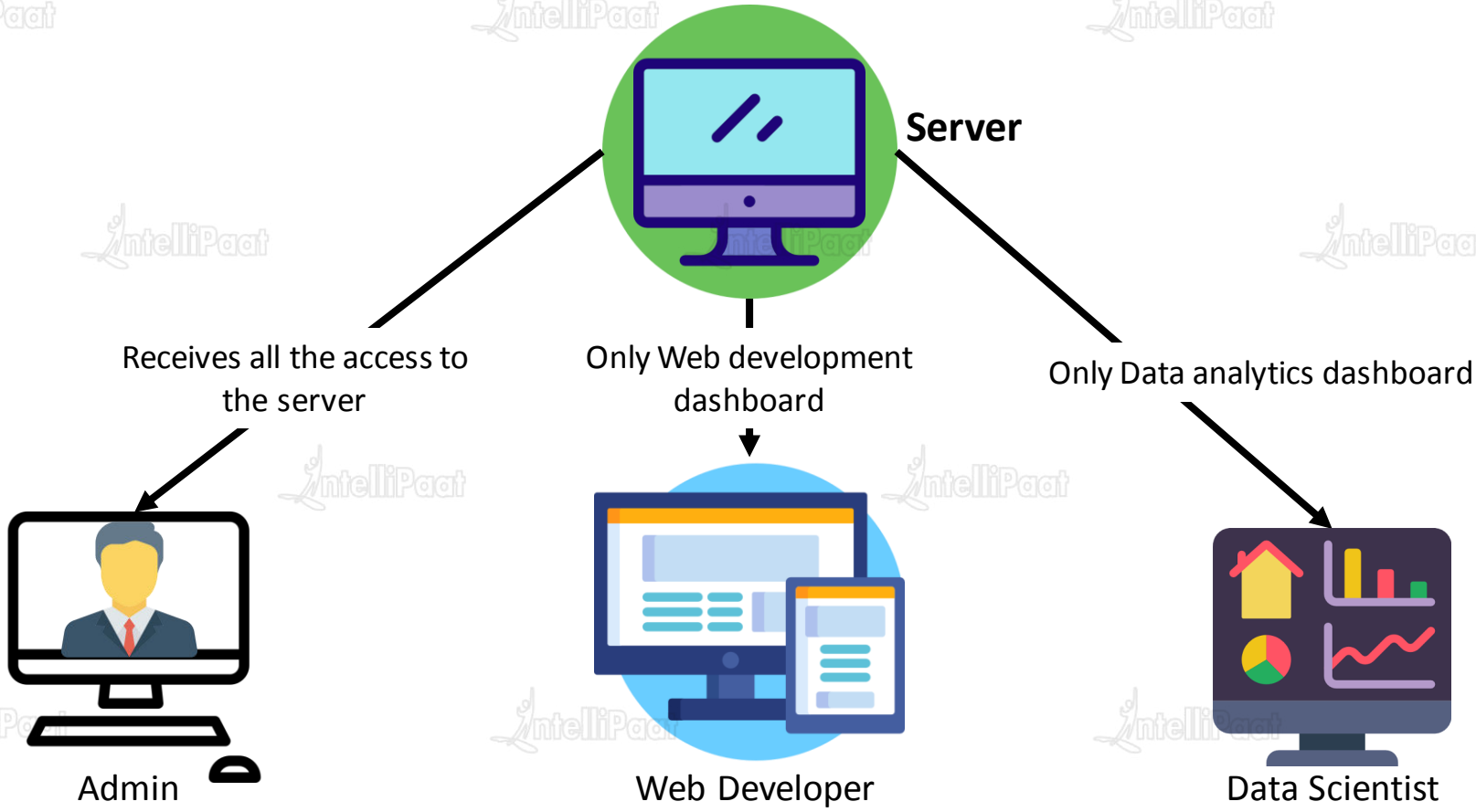
1. Pre-IAM
2. Why Access Management?
3. Amazon Resource Name (ARN)
4. IAM Features
5. Multi-Factor Authentication (MFA)
6. Demo 1: IAM Users & Groups
7. JSON
8. IAM Policies
9. Demo 2: IAM Policies
10. IAM Permissions
11. IAM Roles
12. Demo 3: Roles
13. Identity Federation
14. Pricing
15. Pre-CloudWatch
16. Introduction to CloudWatch
17. Metrics & Namespaces
18. Architecture
19. Dashboard
20. Demo 1: Metrics & Namespaces
21. CloudWatch Alarms
22. Demo 2: Alarm
23. CloudWatch Logs
24. Demo 3: Logs
25. Pricing
26. Design Patterns
27. Quiz

Introduction to IAM

Introduction to IAM

- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.
- You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Why Access Management?

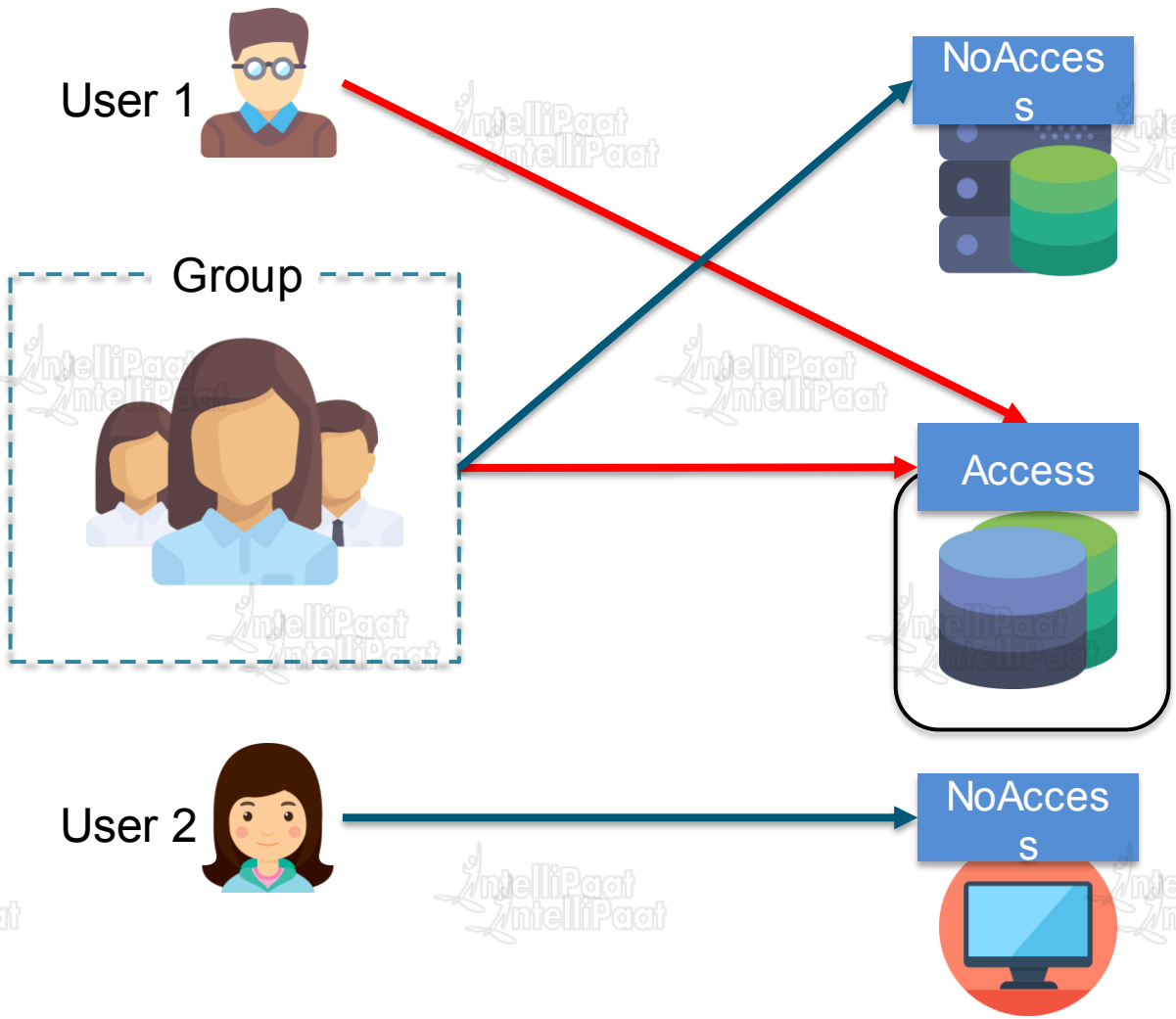


Pre-IAM

Pre-IAM

Users and Groups

- Authentication and Authorization
- Users – A person using AWS services
- Groups – Group of Users
- Permissions – Read, Write or Execute



Amazon Resource Name

Amazon Resource Name

- Amazon Resource Names uniquely identify AWS resources. Every resource in AWS is provided with an ARN.
- ARN Format:

arn:partition:service:region:account-id:resource

arn:partition:service:region:account-id:resourcetype/resource

arn:partition:service:region:account-id:resourcetype:resource

Amazon Resource Name

- EC2
  - ***Instance*** > arn:aws:ec2:region:account-id:instance/instance-id
  - ***AMI*** > arn:aws:ec2:region::image/image-id
  - ***Key-pair*** > arn:aws:ec2:region:account-id:key-pair/key-pair-name
  - ***N/W Interface*** > arn:aws:ec2:region:account-id:network-interface/eni-id
- EBS
  - ***Volume*** > arn:aws:ec2:region:account-id:volume/volume-id
  - ***Snapshot*** > arn:aws:ec2:region:account-id:snapshot/snapshot-id

Amazon Resource Name

VPC > arn:aws:ec2:region:account-id:vpc/vpc-id

Route Table > arn:aws:ec2:region:account-id:route-table/route-table-id

SG > arn:aws:ec2:region:account-id:security-group/security-group-id

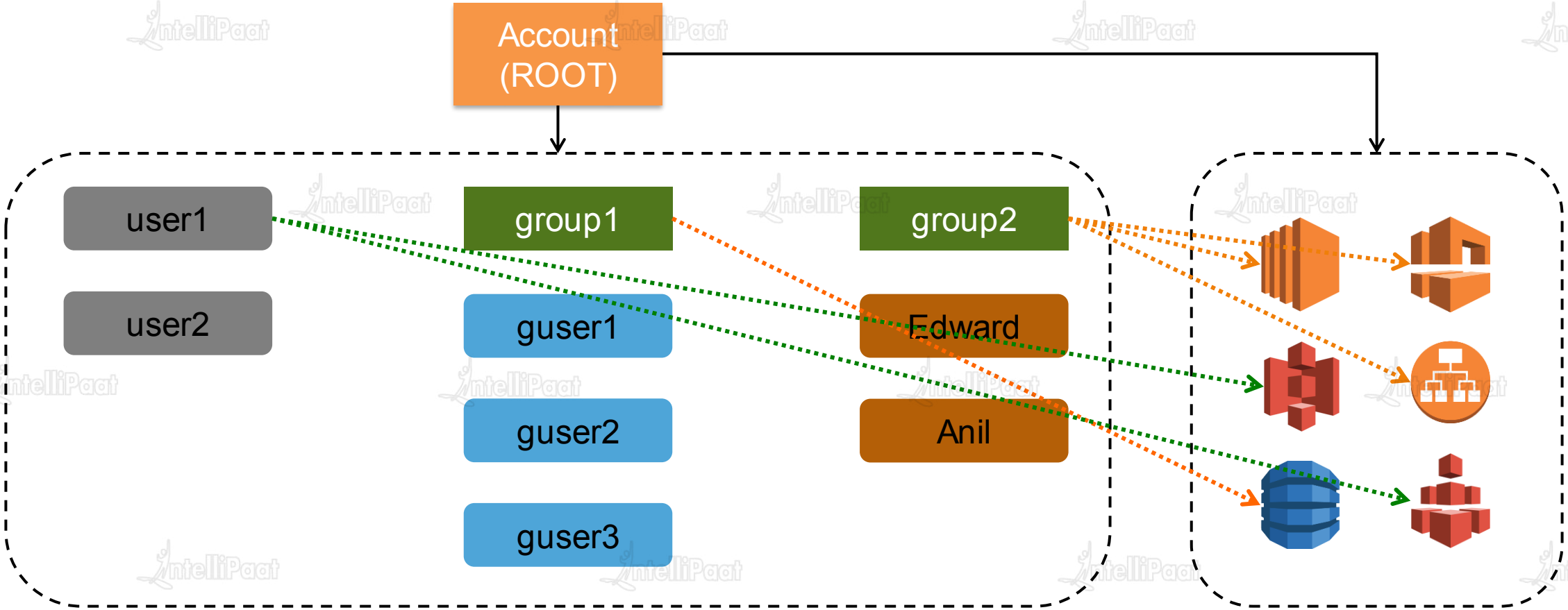
NACL > arn:aws:ec2:region:account-id:network-acl/nacl-id

IGW > arn:aws:ec2:region:account-id:internet-gateway/igw-id

Subnet > arn:aws:ec2:region:account-id:subnet/subnet-id

Peering > arn:aws:ec2:region:account-id:vpc-peering-connection/peering-id

IAM Hierarchy

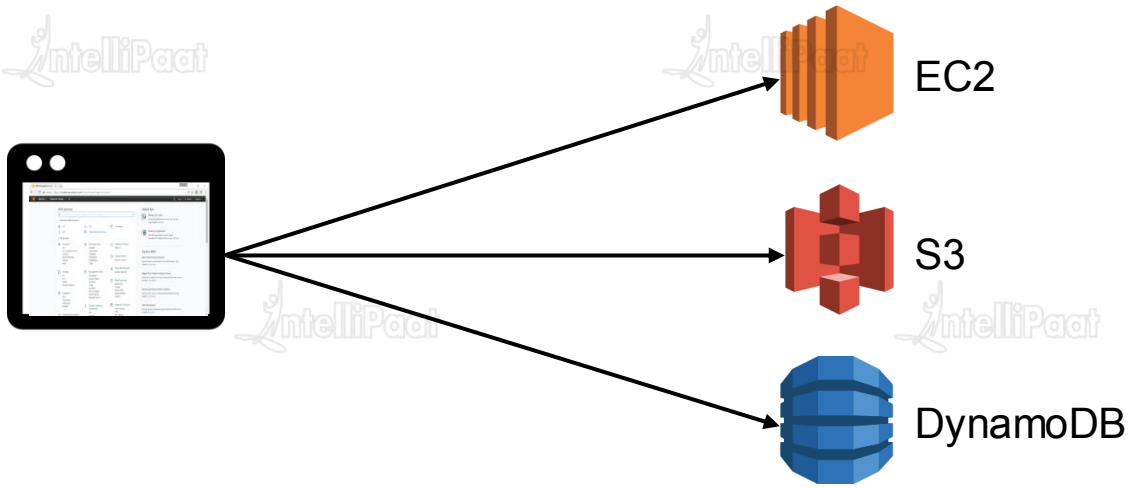


IAM Features

IAM Features

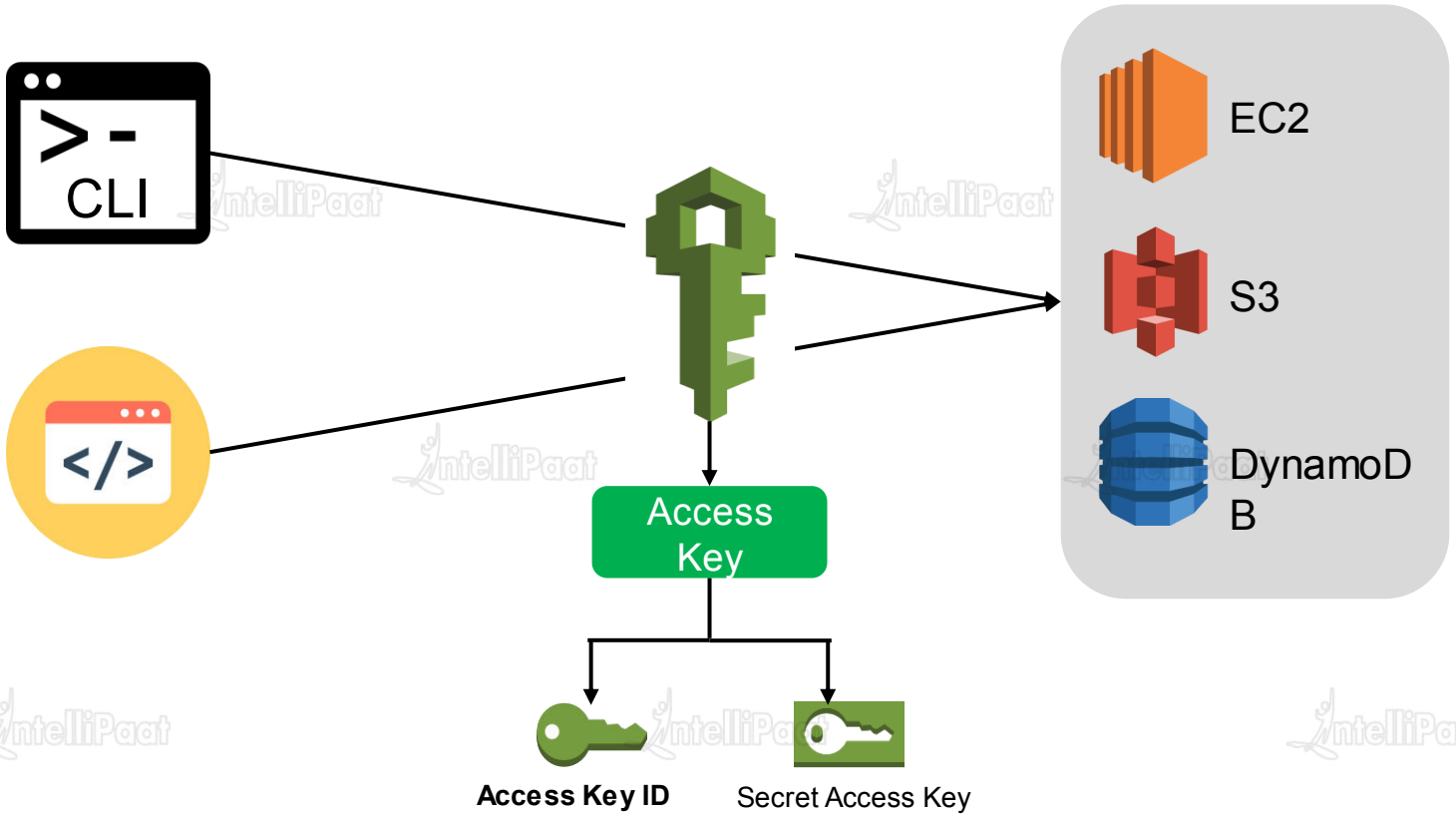
IAM Users

- Represents an entity that is created in AWS, can be a person or service.
- No permissions by default. Nothing is allowed.
- Access requirement
  - Programmatic Access: User needs to make API calls from programs or uses CLI to access AWS resources.
  - Management Console Access: User needs to access AWS resources from management console.

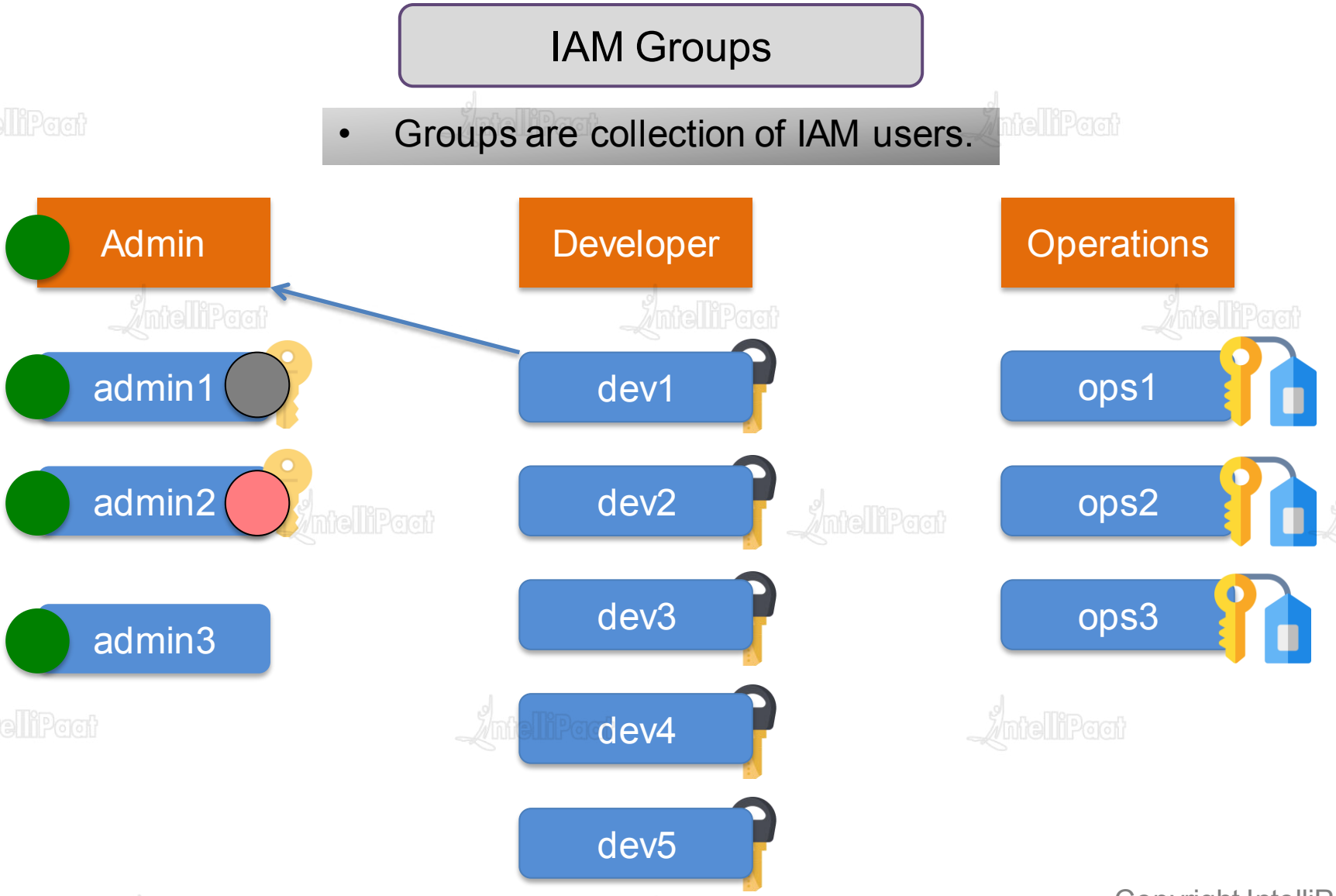


IAM Features

- Access Keys
- Max 2 ACTIVE access keys at a time.
- When disabled access keys cannot be used to make CLI or API calls.



IAM Features



Multi-Factor Authentication

IAM Features

Multi-Factor Authentication

Security Token Based

SMS Based

Demo 1: IAM Users & Groups

Demo 1: IAM Users & Groups

1. Create 2 users using IAM console – admin1, user1.
2. Use “admin1” and “user1” to sign in to the console.
3. Login to the management console using both the users.
4. Create 2 groups – awsfoundation, consolegroup.
5. Add “admin1” to group awsfoundation and “user1” to consolegroup.
6. Create access keys for both the users.
7. Deactivate the access keys.
8. Rotate access keys (only using CLI).
9. Find unused passwords and access keys.
10. Check credential report.
11. Delete all the users and groups.
12. Enable MFA for admin1 user.

The “ROOT” User

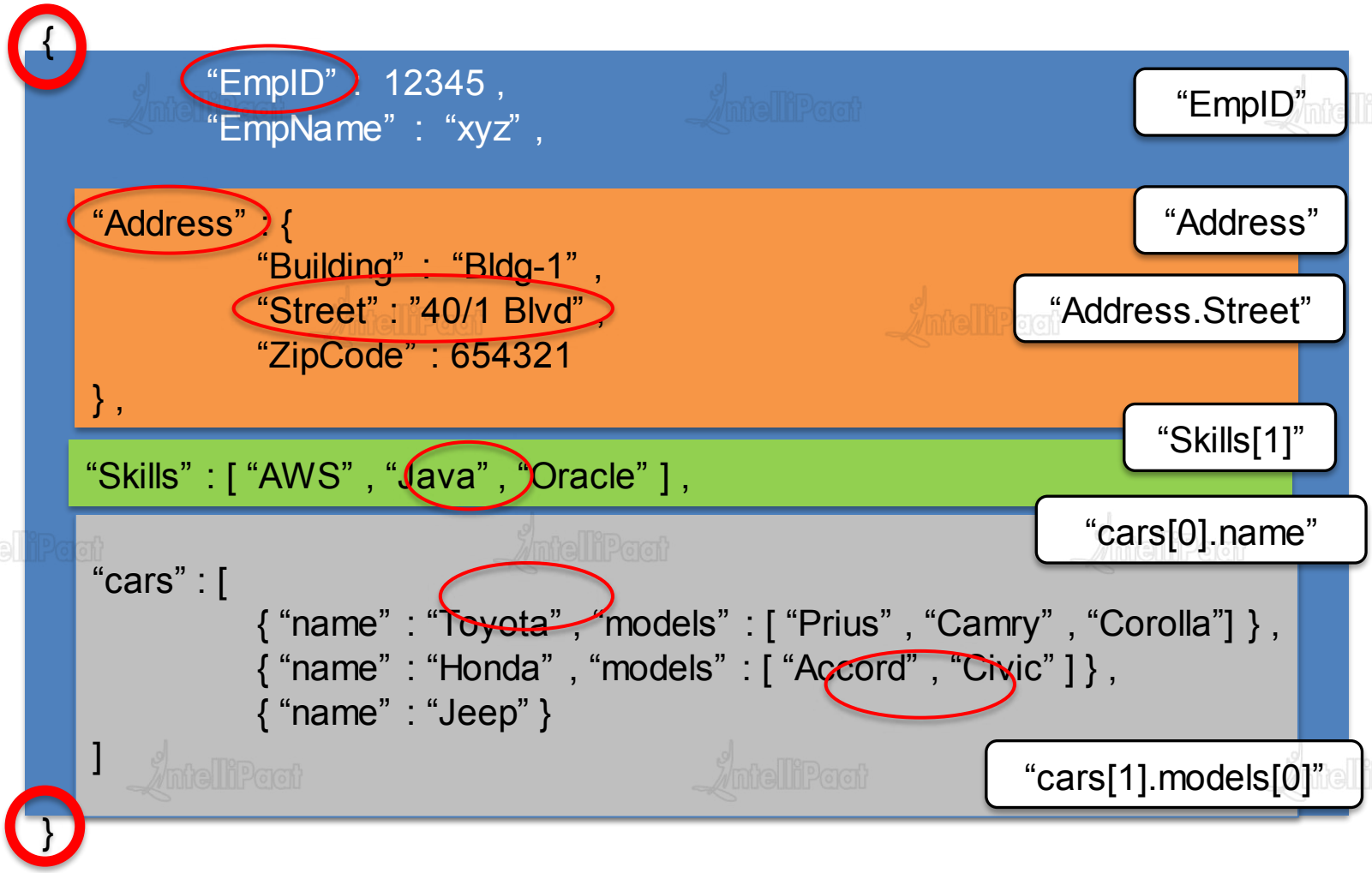
The “ROOT” User

- Root user should not be used at all.
- MFA should be enabled for ROOT user as well.
- ROOT user can also be used for programmatic access.
- Access ID and Secret Access key can be created for ROOT user as well.

JSON

JSON

Introduction to JSON – Java Script Object Notation



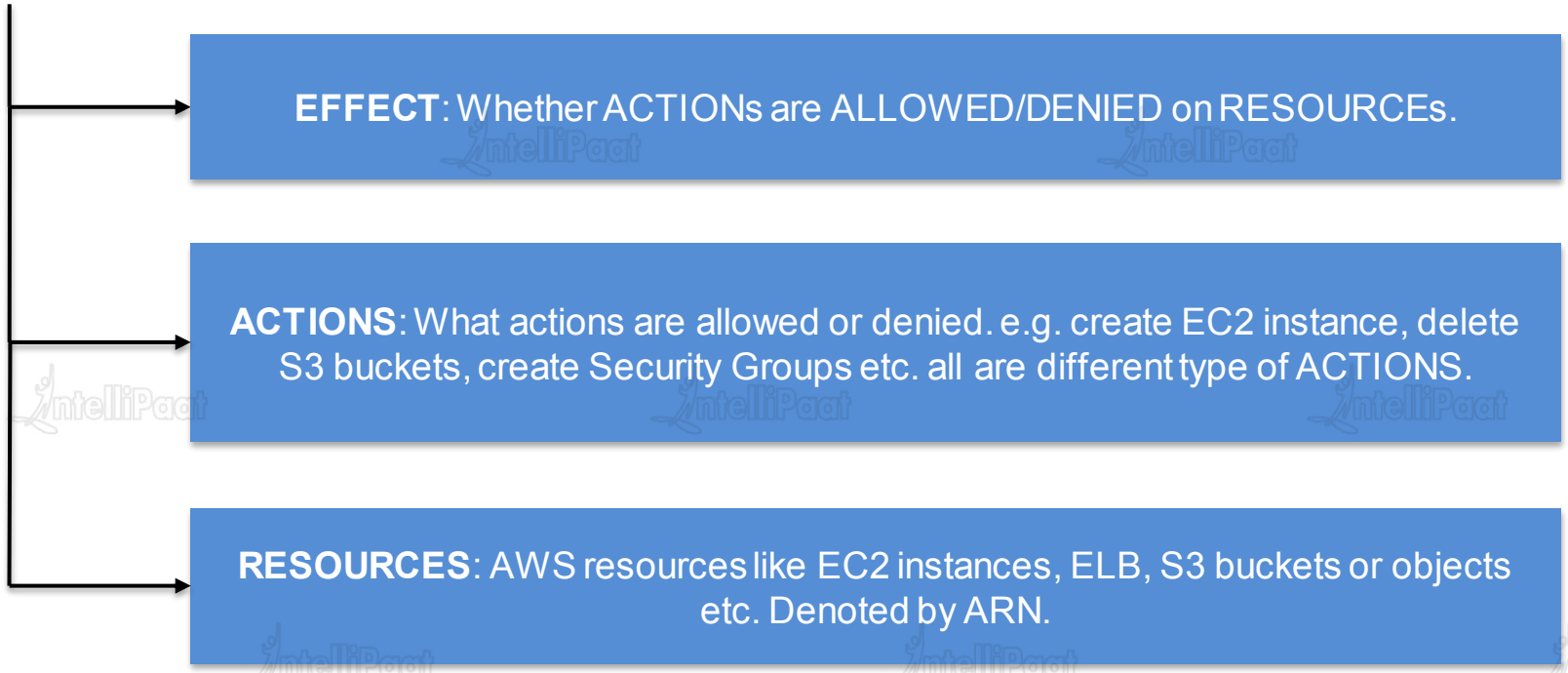
Previous Record

```
{
        "EmpID" : 12345 ,
        "EmpName" : "xyz" ,
        "Address" : {
                "Building" : "Bldg-1" ,
                "Street" : "40/1 Blvd" ,
                "ZipCode" : 654321 ,
        } ,
        "Skills" : [ "AWS", "Java" , "Oracle" ] ,
        "cars" : [
                { "name" : "Toyota" , "models" : [ "Prius" , "Camry" , "Corolla"] } ,
                { "name" : "Honda" , "models" : [ "Accord" , "Civic" ] } ,
                { "name" : "Jeep" }
        ]
}
```

IAM Policies

IAM Policies

- Policies are JSON documents which mention what an user or group can do on AWS resources. It defines the Authorization paradigm for AWS resources.
- Contains 3 components at the least (EAR):



- Policies can be attached to Users or Groups.

IAM Policies

- Resource based policies: when policies are attached to resources.

PRINCIPAL: An entity that can take action on an AWS Resource.

Group

Effect, Action,
Resource : “S3”

S
3

Effect, Action,
Resource : “S3”
Principal : “user-1”

IAM Policies

Policy with a single statement

```
{
 "Version" : "2012-10-17" ,

'Statement " : [
 {   "Effect" : "Allow" ,
     "Action " : "s3:ListBucket" ,
     "Resource" : "arn:aws:s3:::aws-foundation-bucket"
 }
 ]
}
```

Version →
2012-10-17, current version.
2008-10-17, previous version.

IAM Policies

IAM Policies

"Statement" : [{ } , { } , { }]

- Sid : Statement ID.
- Effect : Allow/Deny.
- Principal : ARN of AWS user, account or service which is allowed or denied access to a AWS resource.
- Action : Specific action that is allowed or denied on an AWS resource.
- Resource : ARN of the AWS resource.
- Condition : Condition when a policy is in effect.

- AWS Managed Policies.
- Customer Managed Policies.
- Inline Policies

IAM Policies

Examples

Allow users to access a specific S3 bucket (aws-foundation)

```
{
  "Version": "2012-10-17",
  "Statement": [  // Statement STARTs here
                {
                                "Effect": "Allow",
                                "Action": "s3:ListAllMyBuckets",
                                "Resource": "arn:aws:s3:::*"
                } ,
                {
                                "Effect": "Allow",
                                "Action": [
                                                "s3:ListBucket",
                                                "s3:GetBucketLocation"
                                ],
                                "Resource": "arn:aws:s3:::aws-foundation"
                },
                {
                                "Effect": "Allow",
                                "Action": [
                                                "s3:PutObject",
                                "s3:GetObject",
                                "s3:DeleteObject"
                                ] ,
                                "Resource": "arn:aws:s3:::aws-
                                foundation/*"
                }
  ]  // Statement ENDs here
}
```

Demo 2: IAM Policies

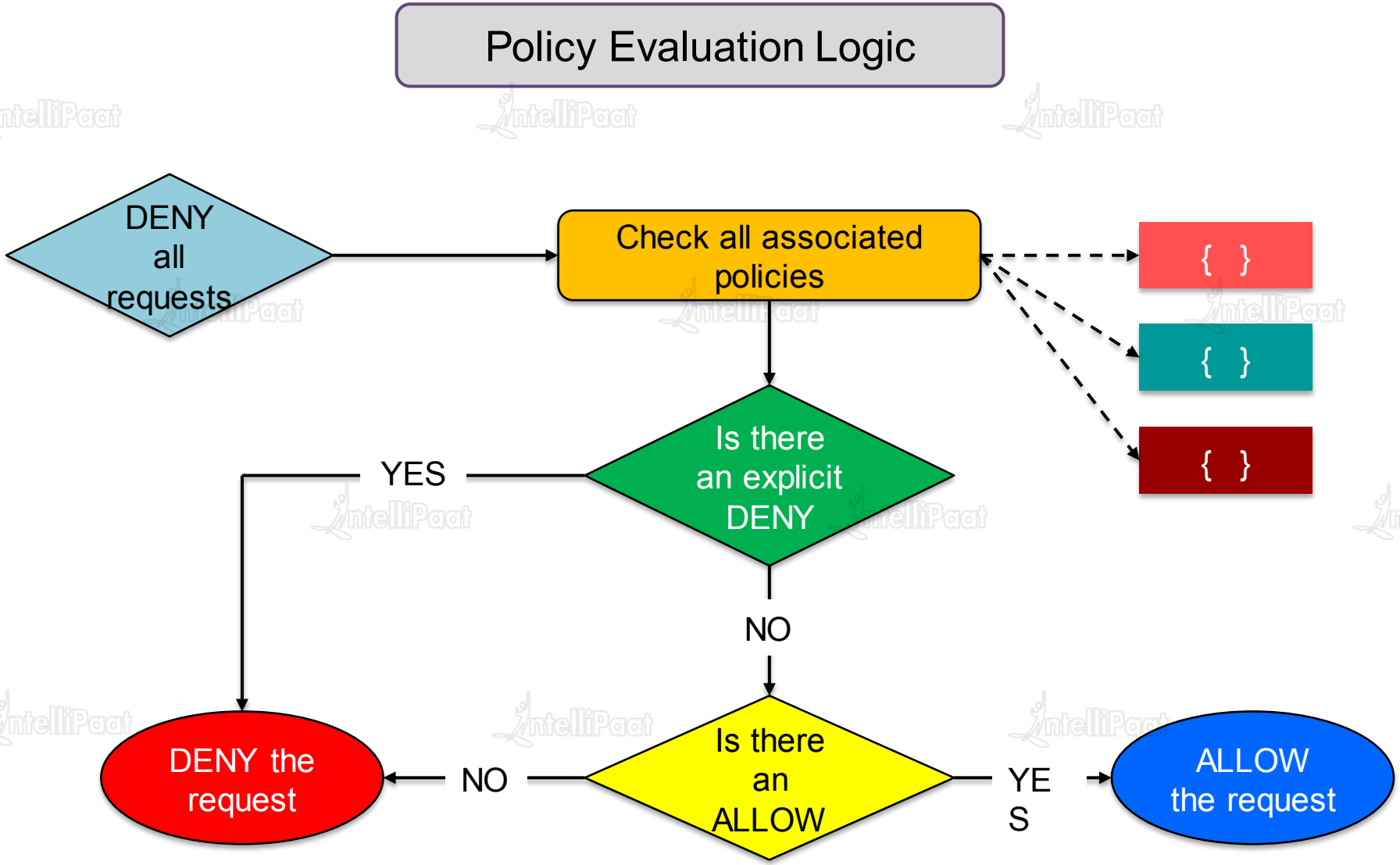
Demo 2: IAM Policies

1) Create a policy with the following
   - » Allow to create EC2 instances.
   - » Allow to list all EC2 instances.
   - » Deny access to terminate EC2 instances.
   - » Allow access to create Classic Load Balancer and launch instances under it.
2) Create policy with the following
   - » Allow access to create VPC, Security Groups, Subnets and Network ACLs.
   - » Allow access to list all objects in a specific S3 bucket.
3) Resource based policy using S3

Demo 2: IAM Policies

1) Select AMI – Need to see the AMIs
2) Select VPC – Need to see all the available VPCs
3) Select SG – Need to see all the available SGs
4) Select Key-Pair
5) Launch the instance

IAM Policy Evaluation Logic



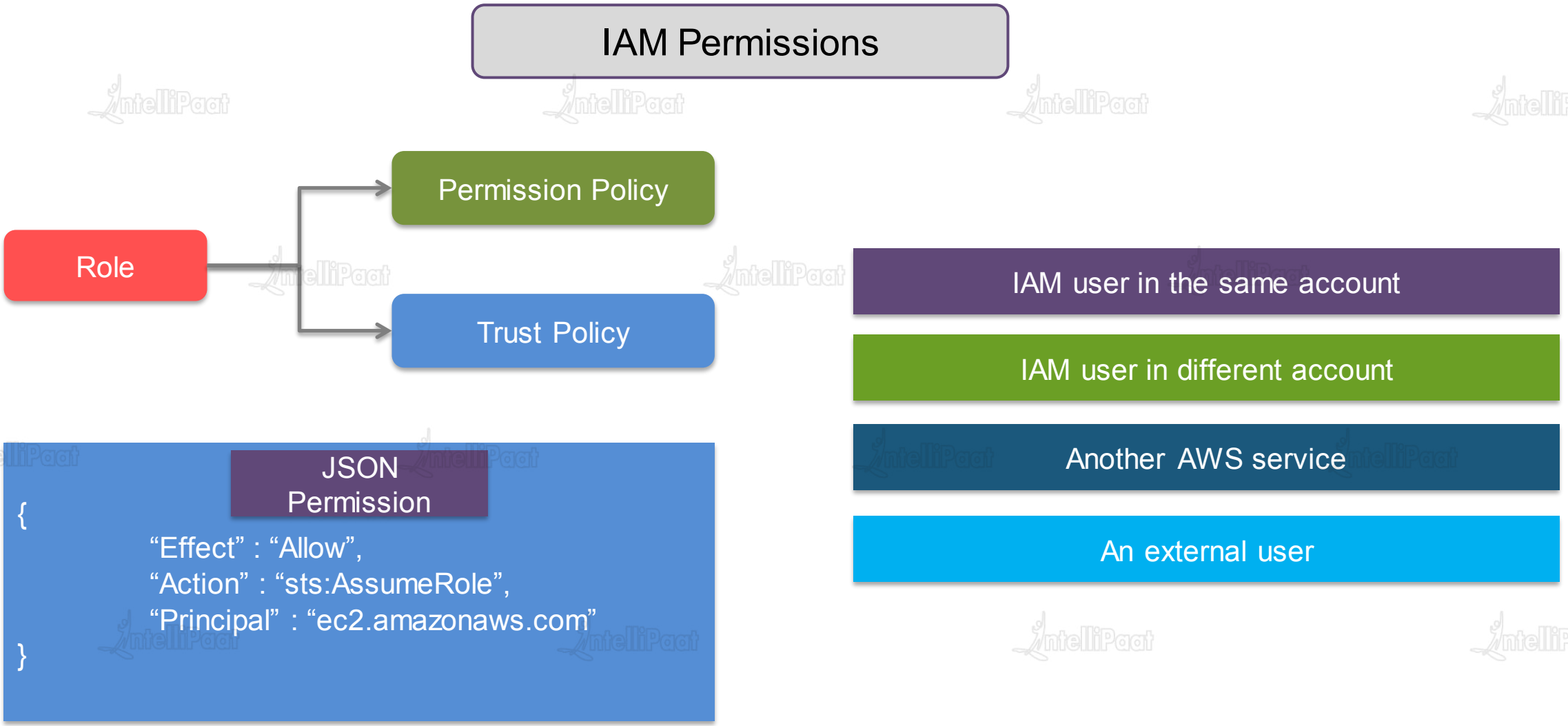
IAM Permissions

IAM Permissions

IAM Permissions

- Permissions are given by attaching policies to users or groups.
- No permission by default for all IAM users.
- AWS account “root” credential.
- Use the policies defined earlier to provide access to users and groups.

IAM Permissions

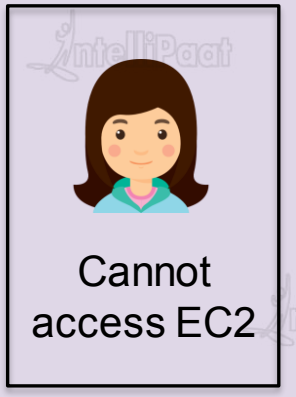


IAM Roles

IAM Roles

IAM Roles

- Role is similar to an user/group which has permissions/policies attached to it.
- Roles are temporary access given to anyone who needs to perform the specific task mentioned in the Role.



- Permissions attached to the users are taken away till the time role is getting used.

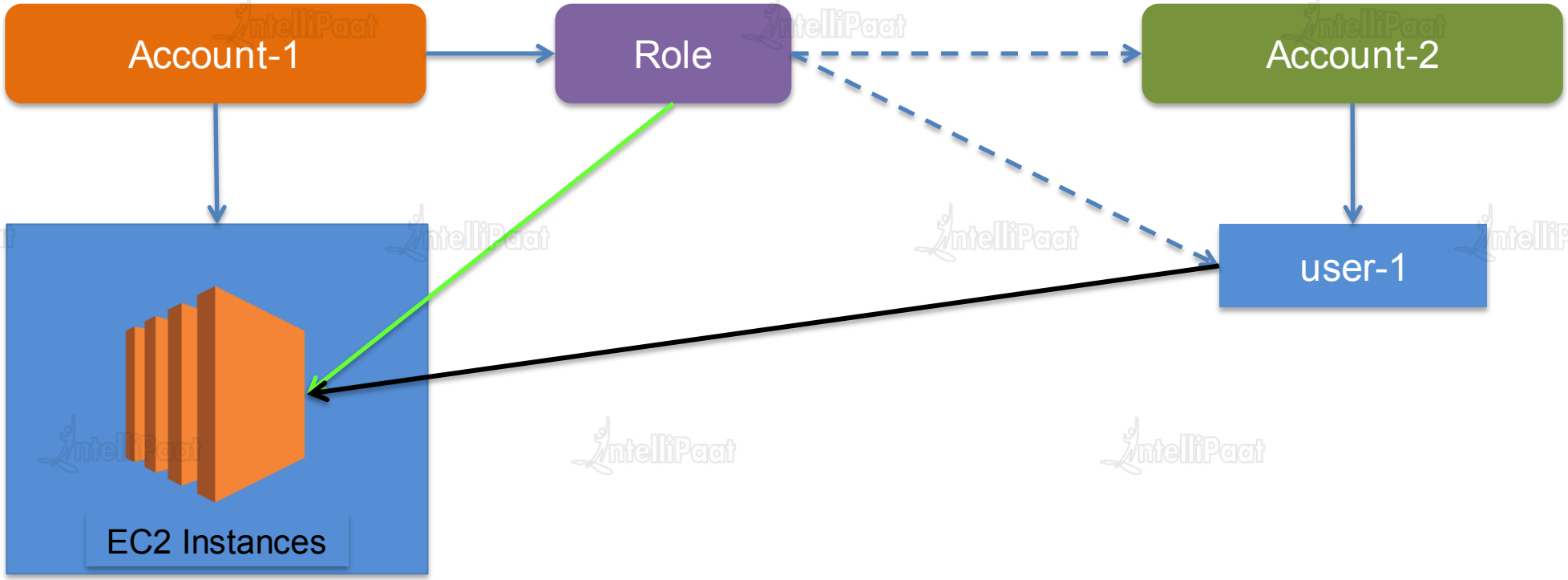
Role: Can access EC2

Role: Can access RDS

IAM Roles

Cross-Account Roles

- Roles and Permissions between Different Accounts and Users.

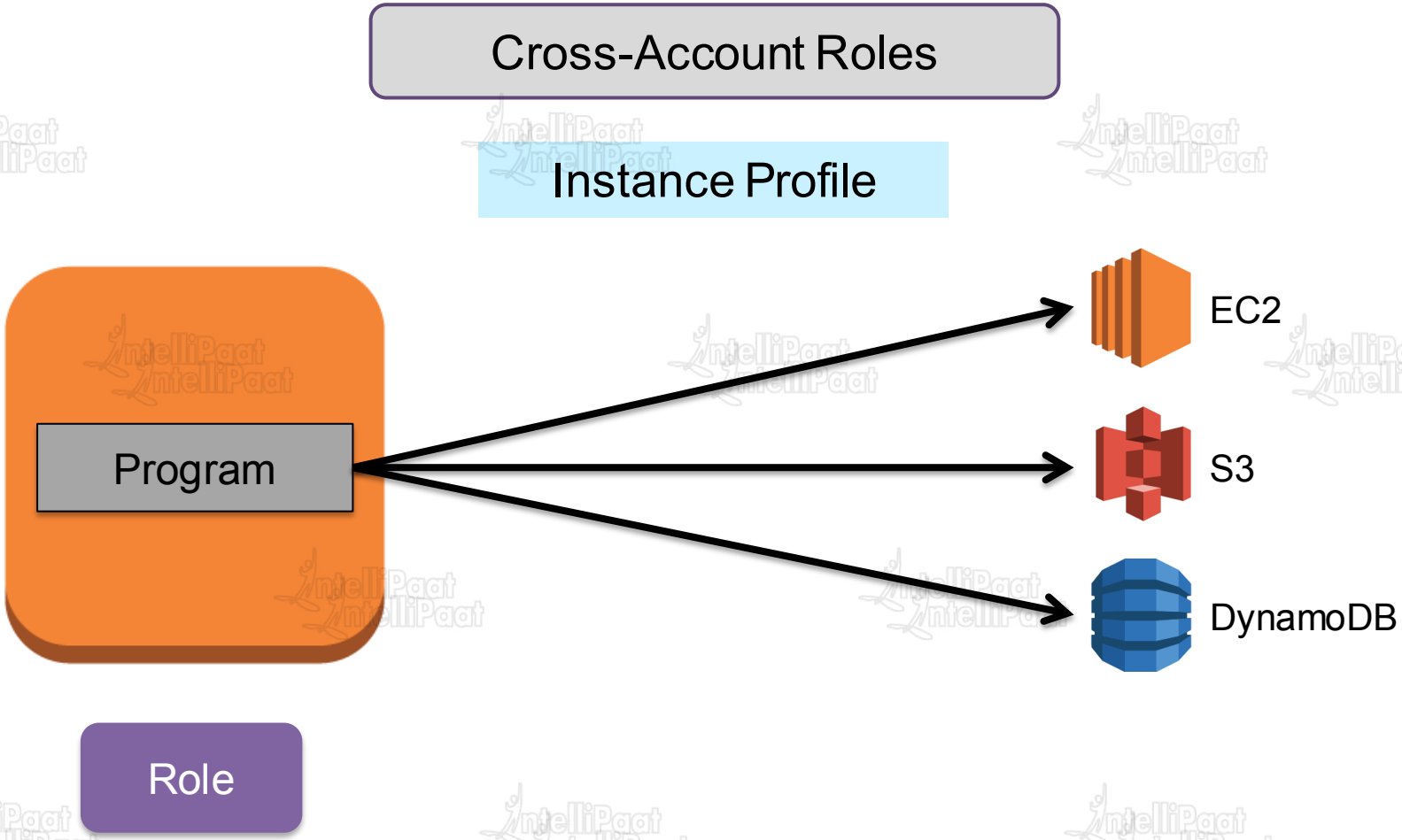


Demo 3: Roles

Demo 3: Roles

1) Create 2 users – s3-user and ec2-user.
2) S3-user should not have any EC2 access.
3) ec2-user should have all access on ec2 instances.
4) Create 2 roles – s3-role & ec2-role.
5) ec2-role should have access to launch EC2 instances and list them.
6) s3-role should have access to the S3 buckets.
7) Make s3-user to assume ec2-role and ec2-user to assume s3-role.

IAM Roles



IAM Roles

Cross-Account Roles

- Identity Federation: AWS resources can be accessed by third party Identity Providers (IdP)
  - *Web*: Facebook, Google, Amazon or any OIDC
  - *SAML2.0*: LDAP or Microsoft AD

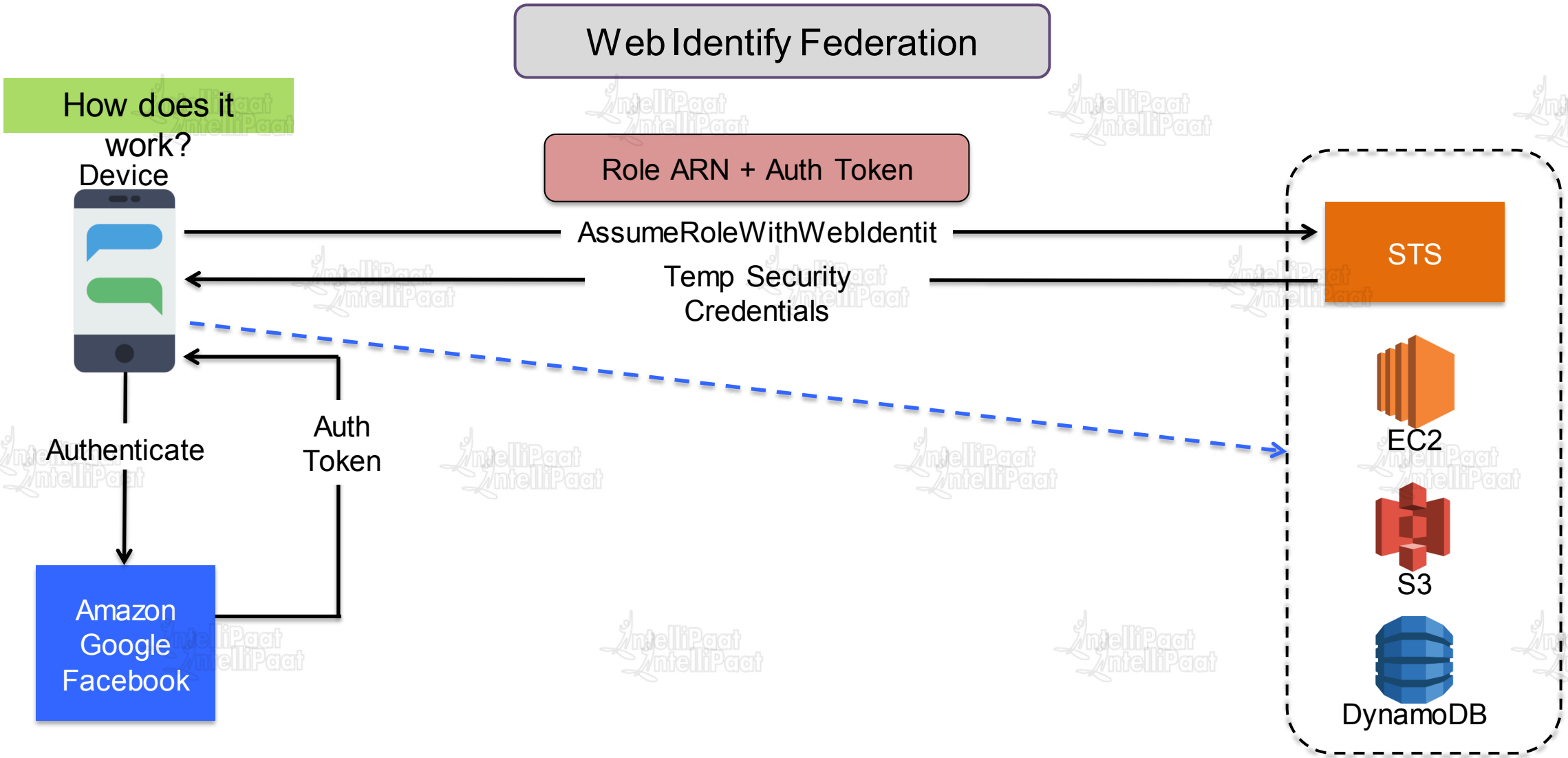
- Steps (Web Identity Federation)
  - Sign up as developer in Facebook or Google or Amazon account.
  - Create an Identity Provider in IAM.
  - Create Role with Trust and Permission Policy
  - In Trust Policy Principal should be the Web IdP
  - Cognito can be used as Identity Broker.

```
"Principal" : { "Federated" : "www.amazon.com" }
"Principal" : { "Federated" : "graph.facebook.com" }
"Principal" : { "Federated" : "accounts.google.com" }
```

```
"Action" : "sts:AssumeRoleWithWebIdentity"
```

Identity Federations

IAM Federations



IAM Federations

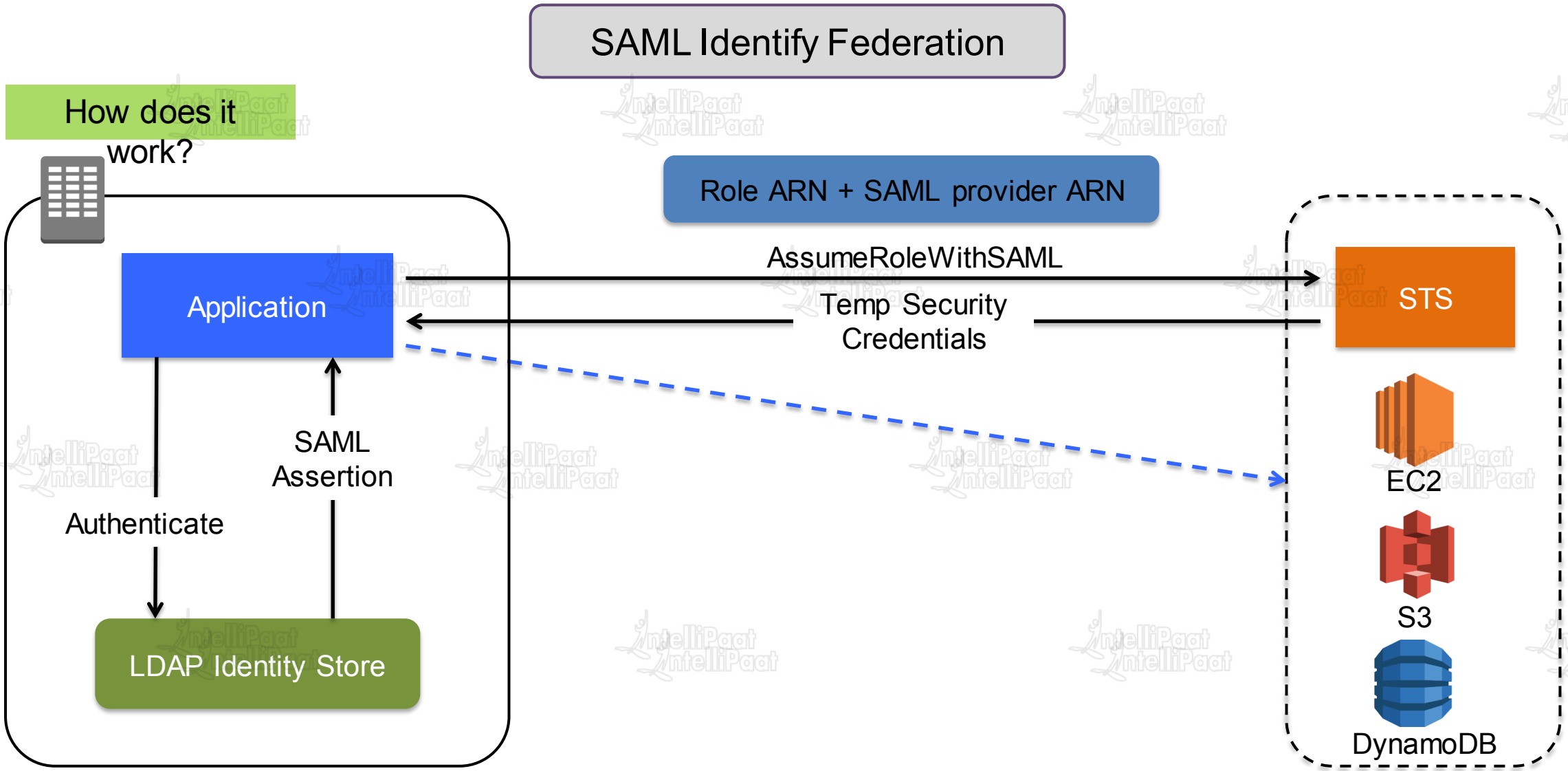
SAML Identify Federation

- Steps (SAML Federation)
  - Register AWS with Corporate IdP (LDAP).
  - That will generate a Metadata XML.
  - Create a SAML identity provider with the SAML metadata.
  - Create Roles.
  - These roles should be mapped with Organization's assertions.

"Principal" : { "AWS" : "ARN of the SAML provider" }

"Action" : "sts:AssumeRoleWithSAML"

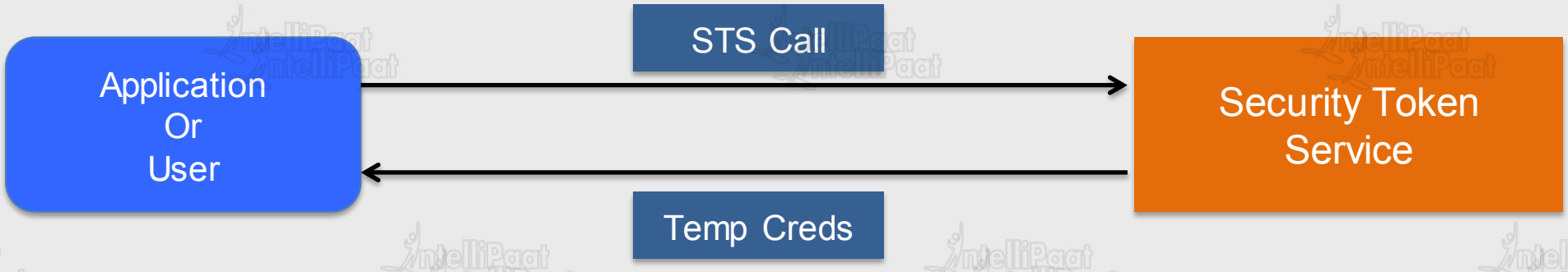
IAM Federations



Security Token Service

Temporary Security Credentials & STS

- STS (Security Token Service) can be used to get temporary security credentials.
  - Temporary Access Key ID, Secret Access Key and Security Token



- STS Calls.
  - "AssumeRole": ARN of the Role, Duration (15 mins to 1 hour (Default))
  - "AssumeRoleWithWebIdentity": ARN of the Role, Auth Token, Duration (15 mins to 1 hour (Default))
  - "AssumeRoleWithSAML" : ARN of the Role, ARN of the SAML provider created in IAM, SAML assertion, Duration (15 min to 1 hour (Default)
  - "GetFederationToken"
  - "GetSessionToken"

Pricing

Pricing

Entirely Free!!

Summary

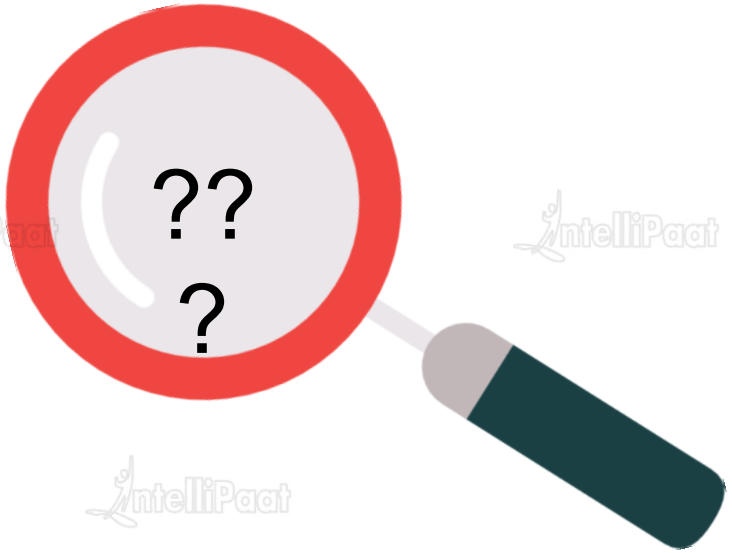
- Authorization & Authentication.
- Amazon Resource Name (ARN), IAM Hierarchy.
- IAM Users, Groups and Roles.
- Multi-Factor Authentication.
- Policy Evaluation.
- IAM Roles
  - Roles in the same account
  - Cross-account Roles
- Instance Profile
- Identity Federation – Web (OIDC) and SAML.

Pre-CloudWatch

Pre-CloudWatch

Monitoring

Before CloudWatch, Monitoring was taking place by collecting related data and analyzing manually.

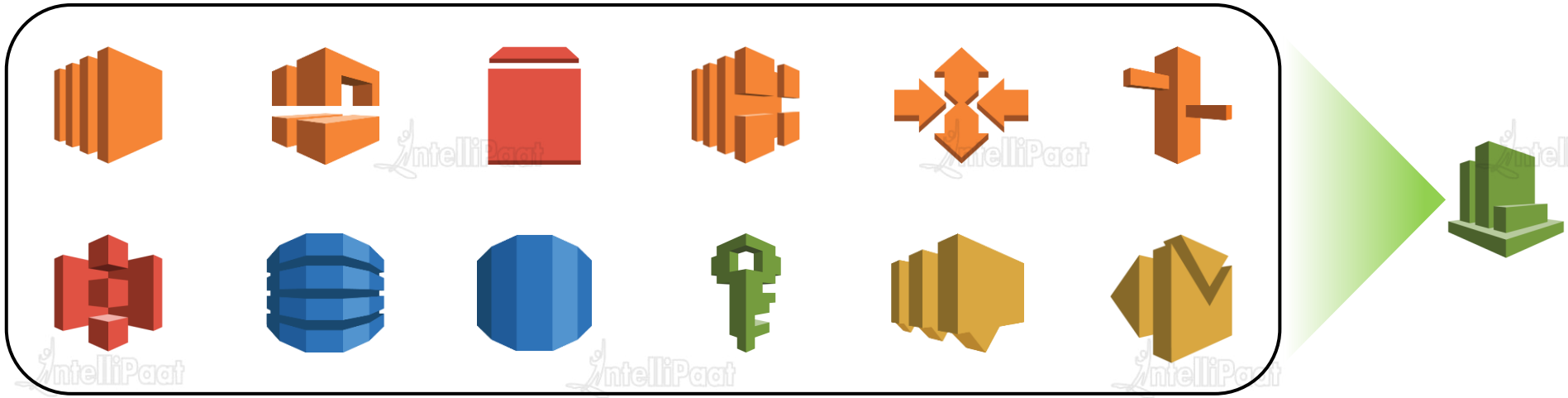


Introduction to CloudWatch

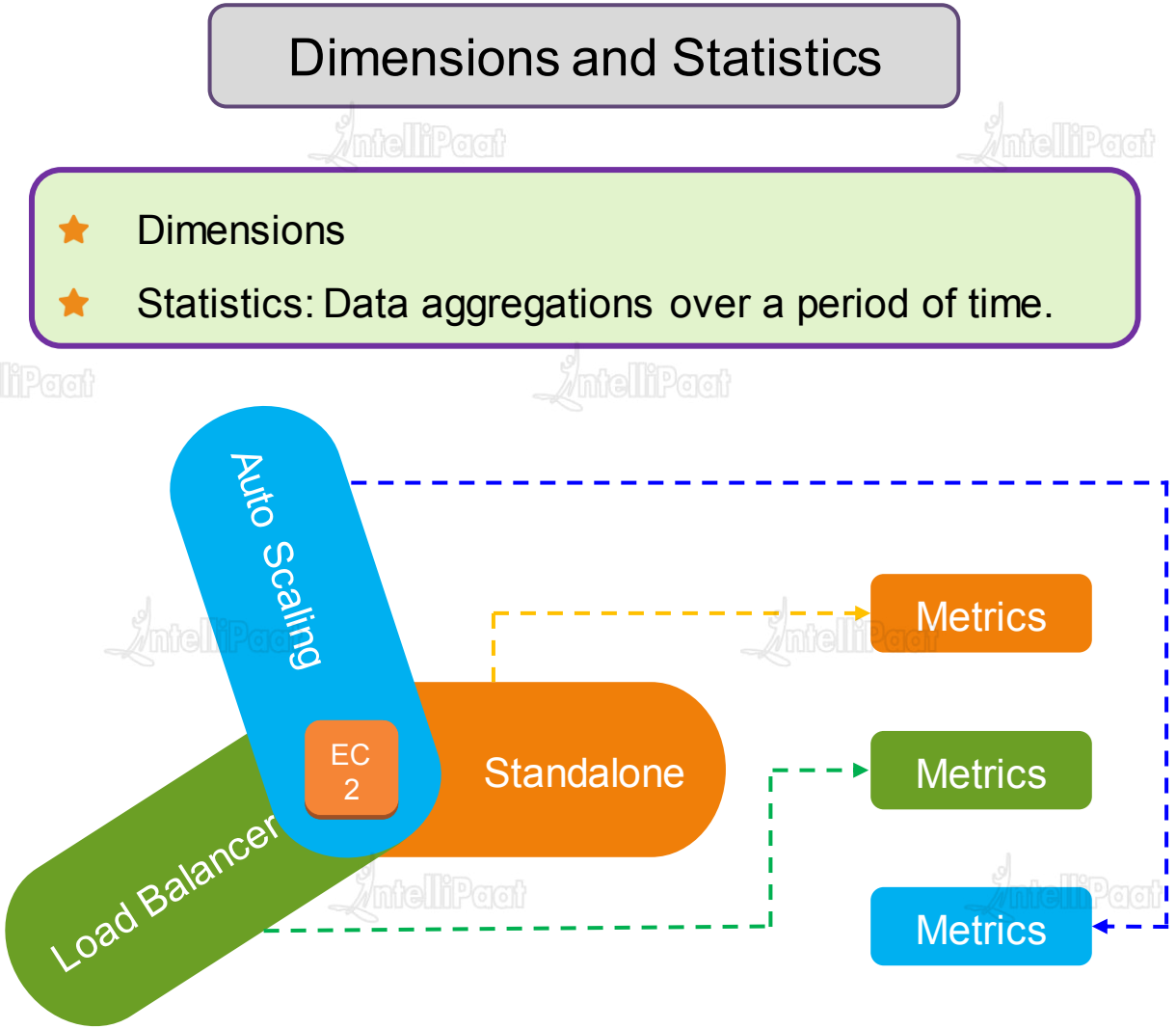
Introduction to CloudWatch

CloudWatch Monitoring

- Monitors all AWS resources provisioned and deployed.
- Sends notifications if anything goes wrong.
- **Following services are used in conjunction with CloudWatch:**



Dimensions and Statistics

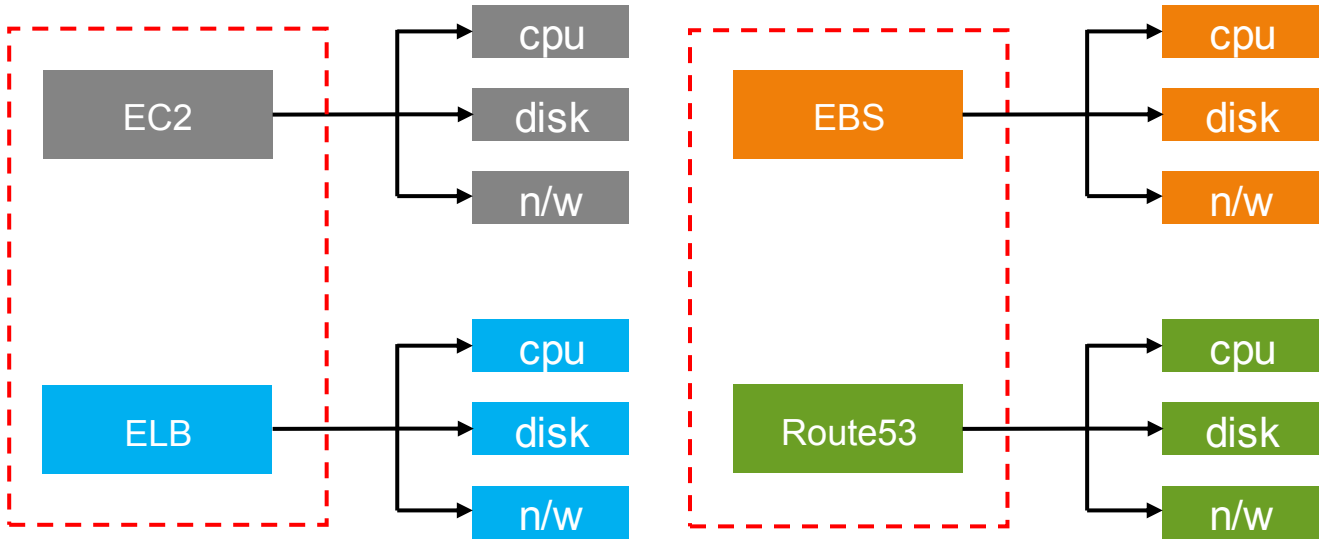


CloudWatch Metrics and Namespaces

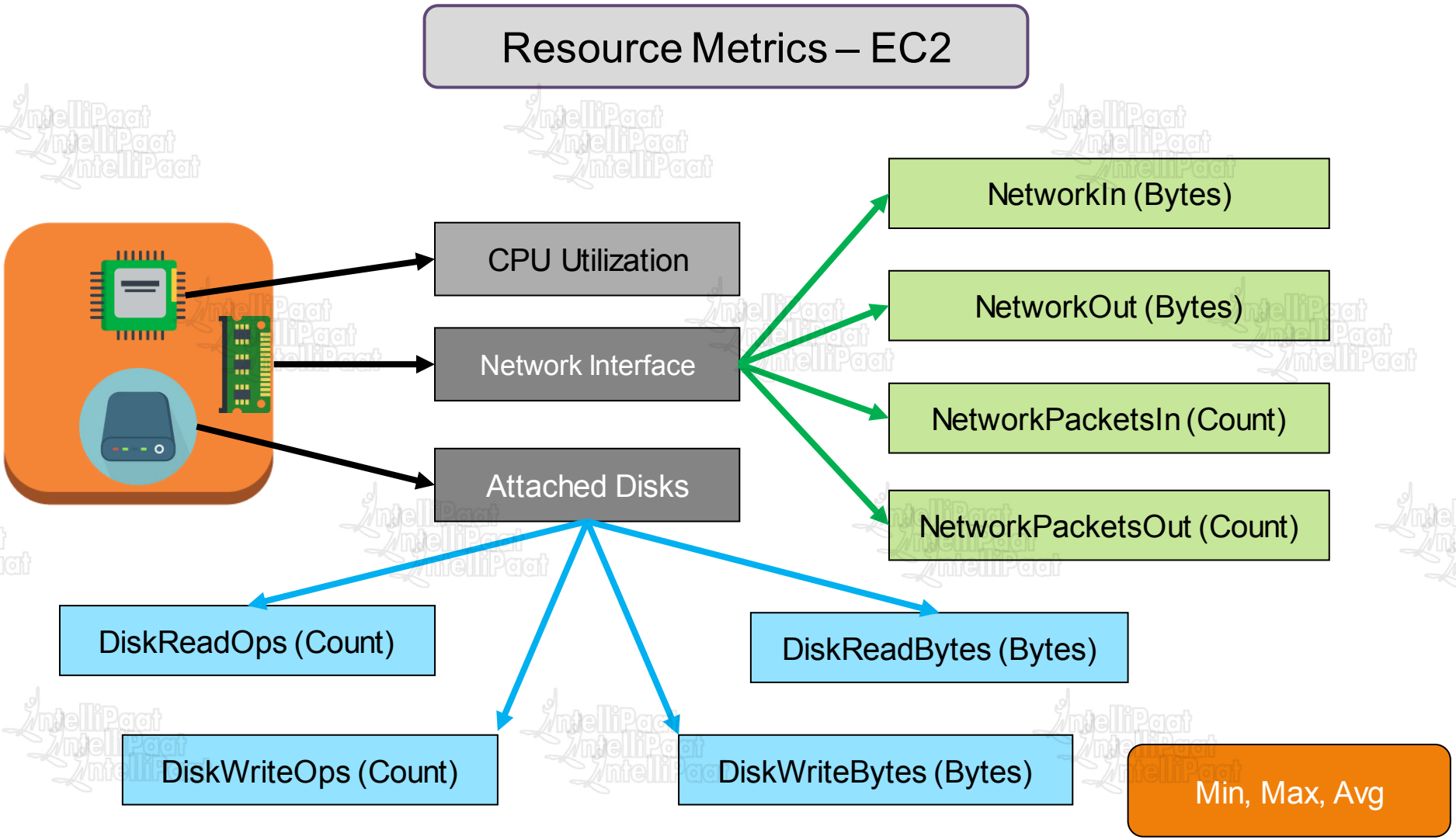
CloudWatch Metrics and Namespaces

Metric and Namespaces

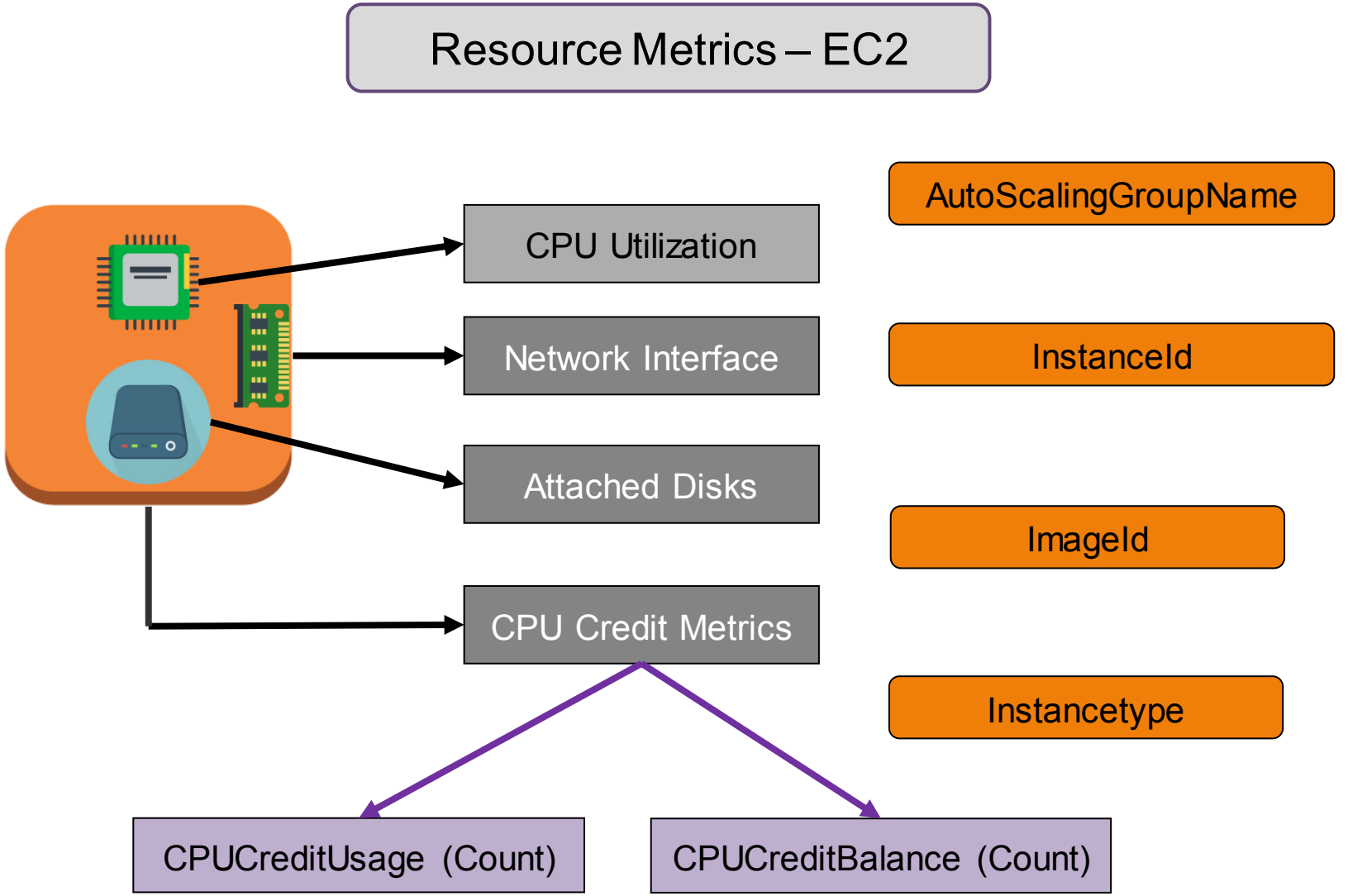
- Metrics are fundamental to CloudWatch monitoring.
- Individual data points which are monitored, all actions are based on metrics. e.g. CPU Utilization percentage.
- All AWS services send metrics to CloudWatch by default.



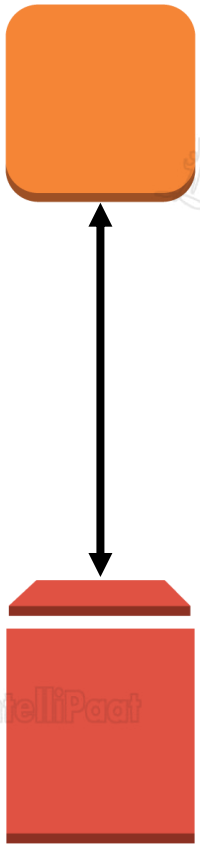
CloudWatch Metrics and Namespaces



CloudWatch Metrics and Namespaces



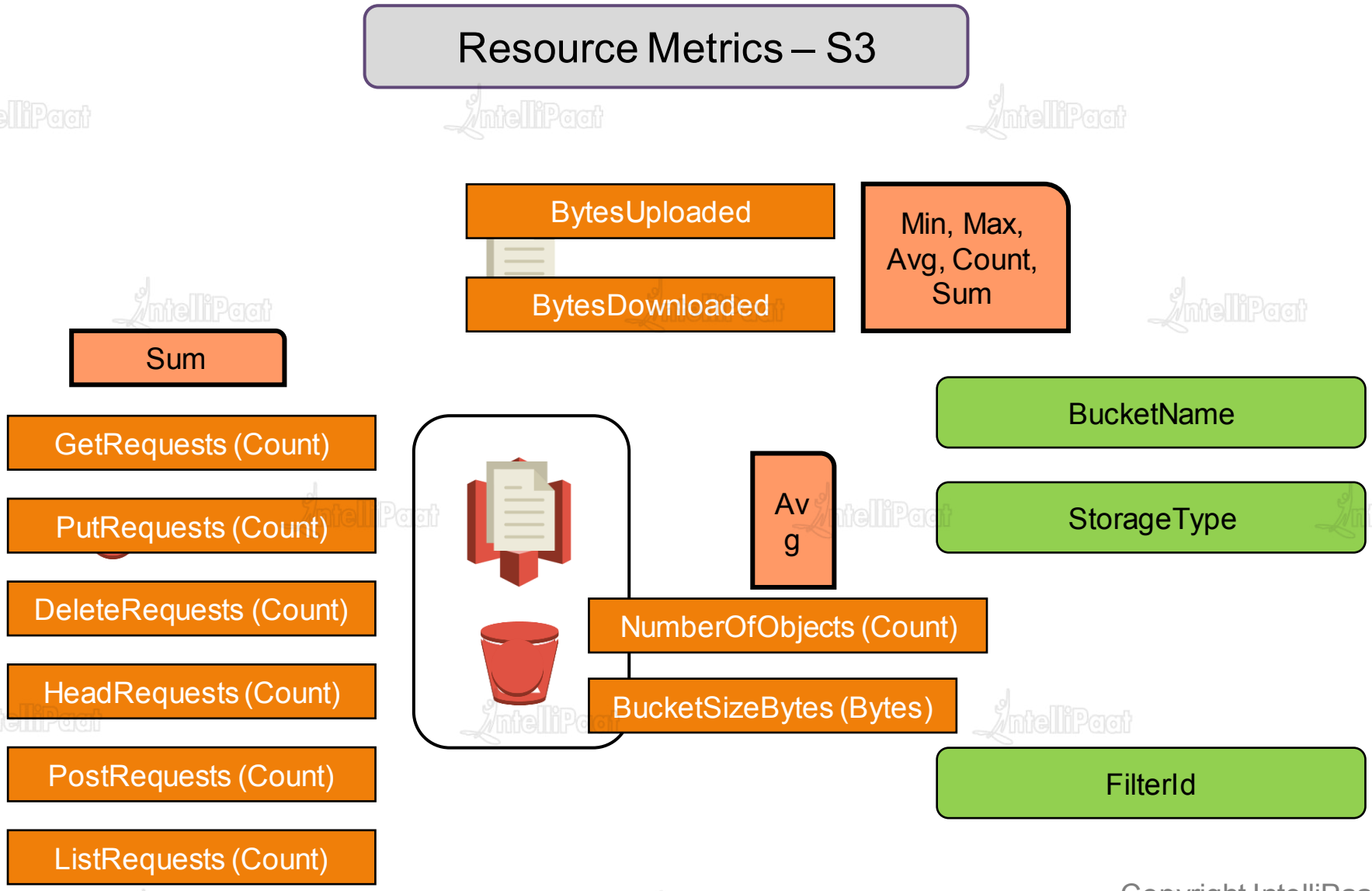
CloudWatch Metrics and Namespaces



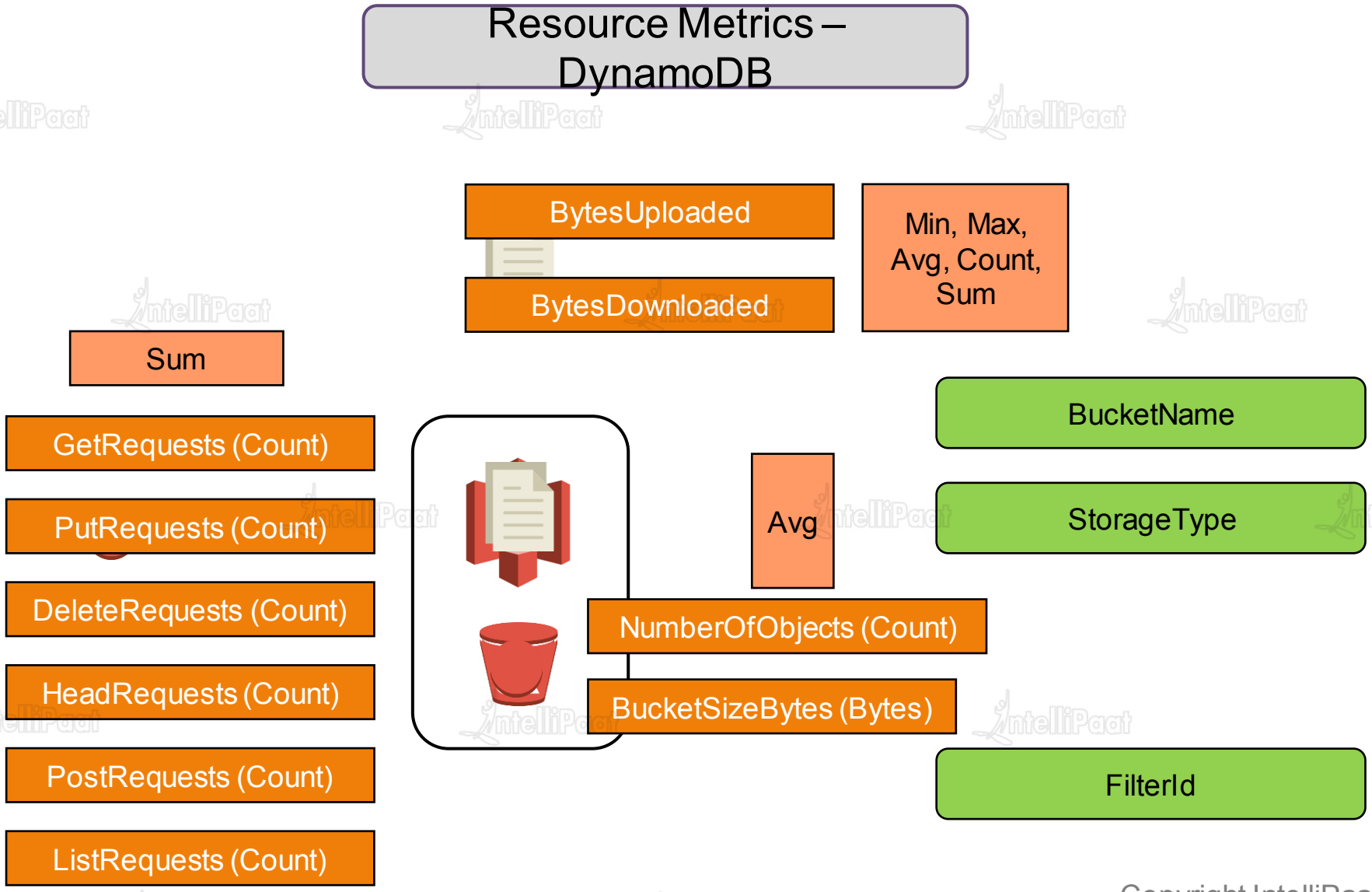
Resource Metrics – EBS

Metrics	Unit	Statistics
VolumeReadBytes	Bytes	Sum, Avg, Count
VolumeWriteBytes	Bytes	Sum, Avg, Count
VolumeReadOps	Count	
VolumeWriteOps	Count	
VolumeTotalReadTime	Seconds	
VolumeTotalWriteTime	Seconds	
VolumeIdleTime	Seconds	
VolumeQueueLength	Count	
VolumeThroughputPercentage	Percent	
VolumeConsumedReadWriteOps	Count	
BurstBalance	Percent	

CloudWatch Metrics and Namespaces



CloudWatch Metrics and Namespaces



CloudWatch Metrics and Namespaces

Resource Metrics – DynamoDB

GlobalSecondaryIndexName

StreamLabel

TableName

PutItem
DeleteItem
UpdateItem
GetItem
BatchGetItem
Scan
Query
BatchWriteItem

Table

Item

Attributes

Min, Max, Avg, Count, Sum

GSI/LSI

ConsumedReadCapacityUnits

ProvisionedReadCapacityUnits

ConsumedWriteCapacityUnits

ProvisionedWriteCapacityUnits

OnlineIndexConsumedWriteCapacity

ReadThrottleEvents

OnlineIndexPercentageProgress

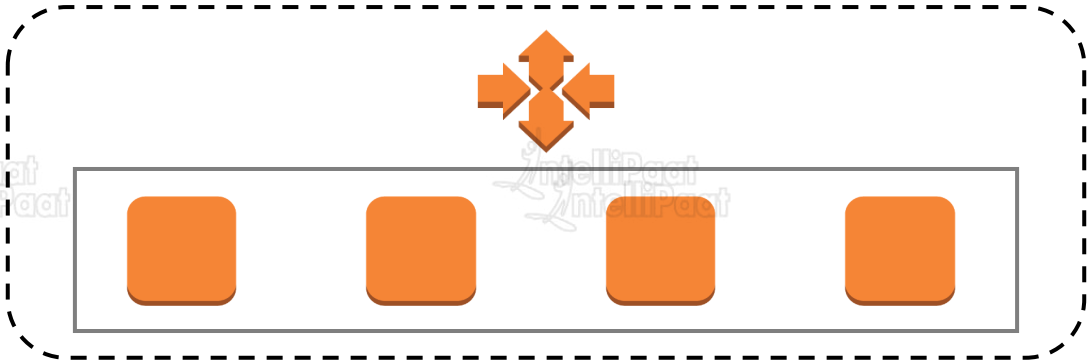
WriteThrottleEvents

OnlineIndexThrottleEvents

ThrottledRequests

CloudWatch Metrics and Namespaces

Resource Metrics – AS



GroupMinSize

GroupMaxSize

GroupDesiredCapacity

GroupInServiceInstances

GroupPendingInstances

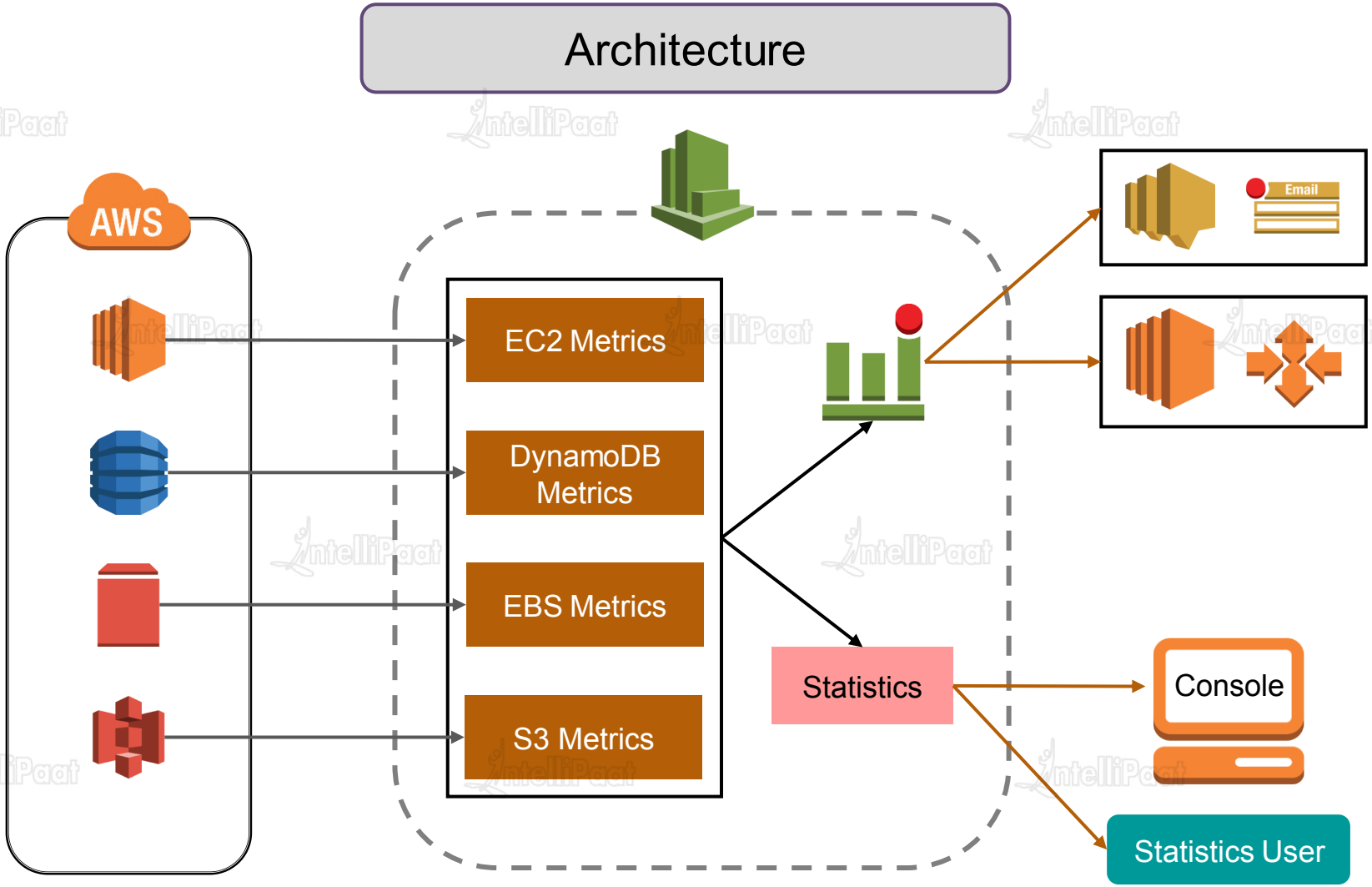
GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

CloudWatch Architecture

CloudWatch Architecture



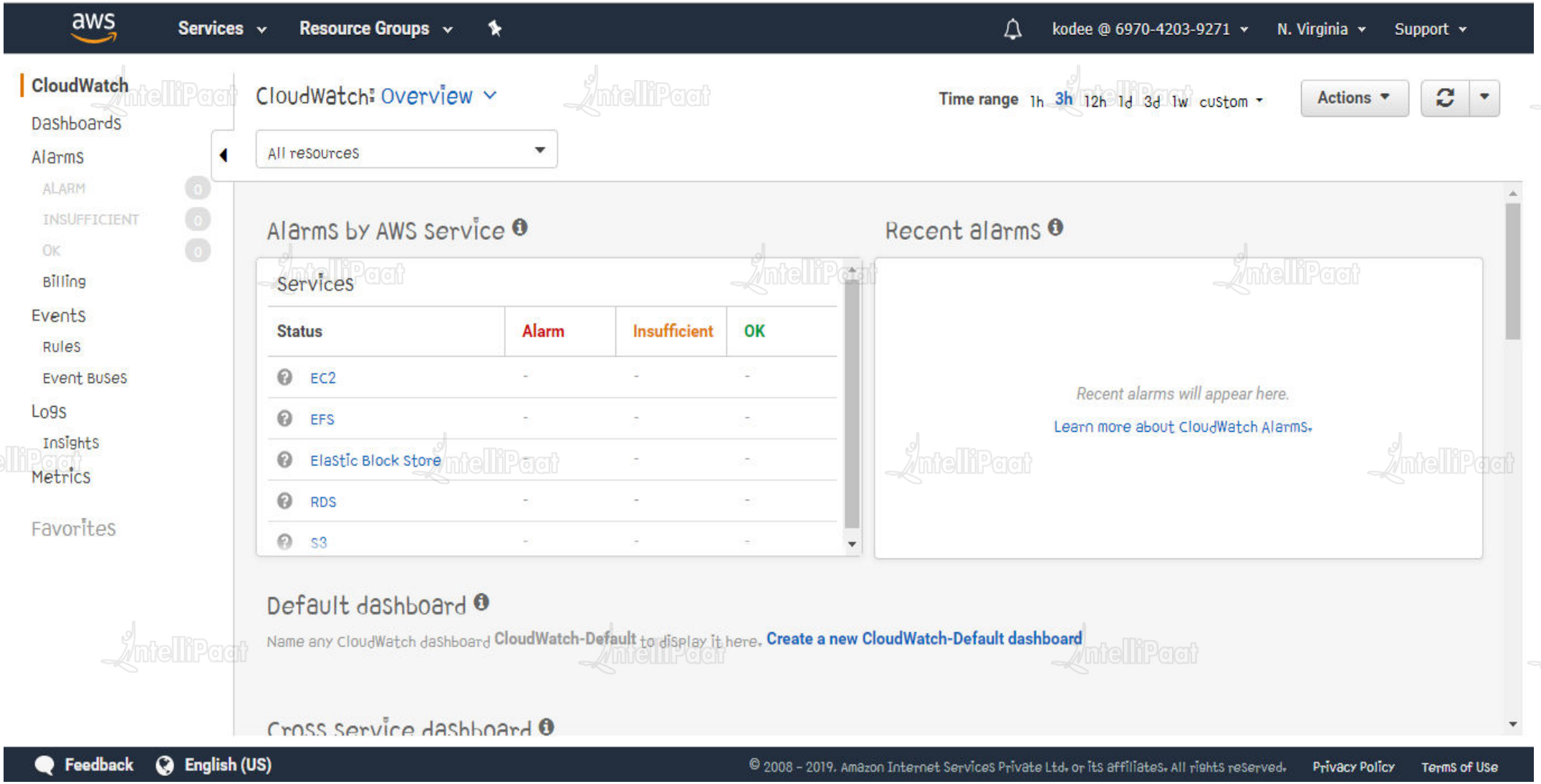
CloudWatch Dashboard

CloudWatch Dashboard

Dashboards

Dashboards are pages in the console which can be used to put all the important statistics deemed important at one place.

CloudWatch Dashboard



Demo 1: Dashboard & Metric Stats

Demo 1: Dashboard & Metric stats

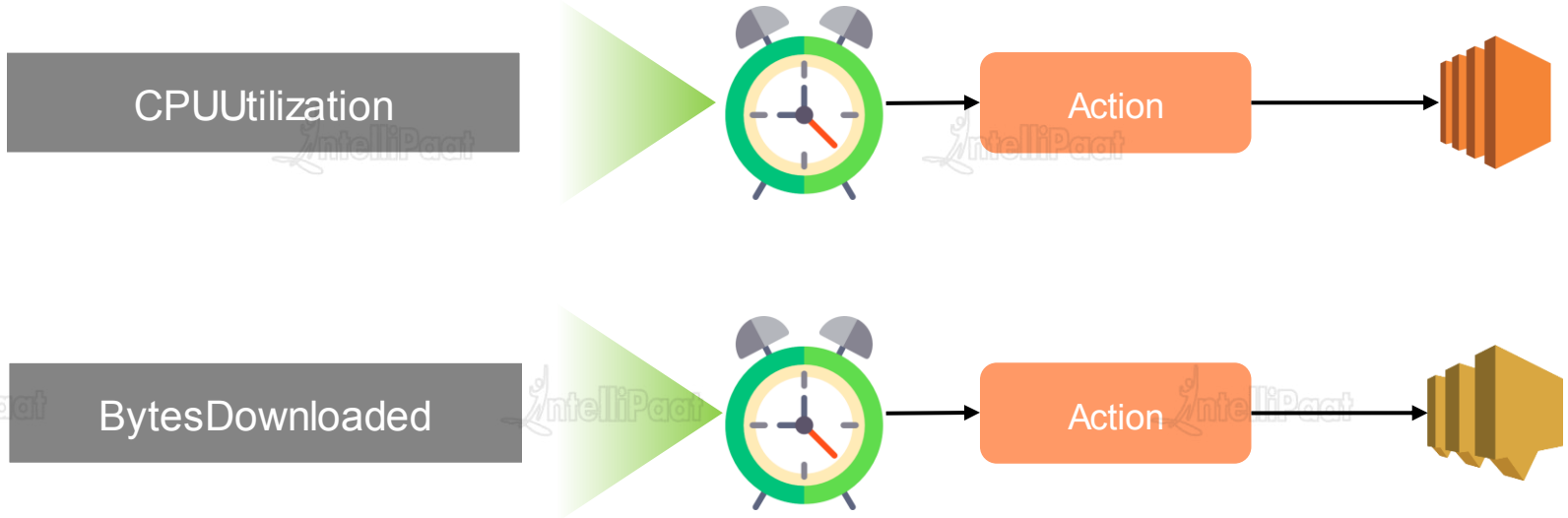
- Create a Dashboard – AWS-foundation-dashboard.
- Add widgets for EC2, ELB and S3.
- Graph Options.
- Graphed Metrics.
- Edit a widget.
- Rename the graph.
- View Statistics for metrics for a specific resource.

CloudWatch Alarm

CloudWatch Alarm

Alarm

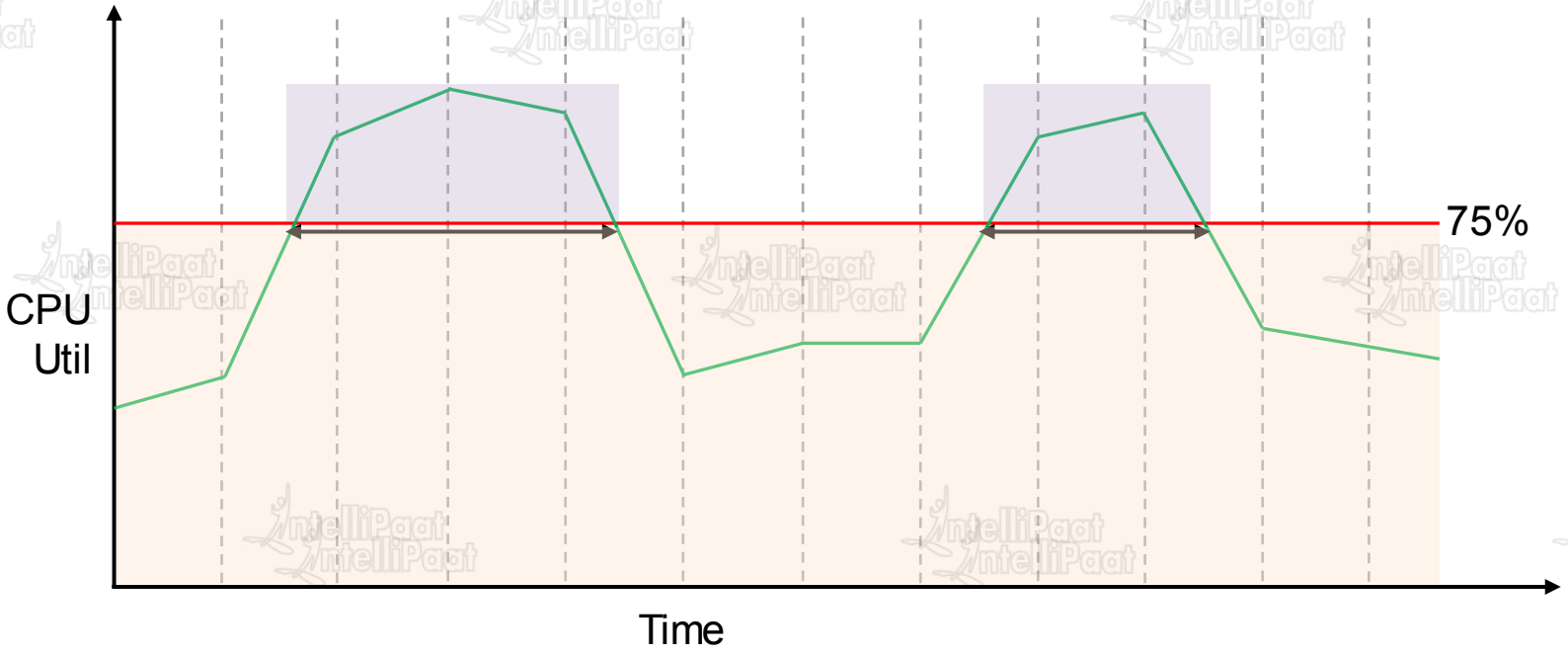
- Alarms watch over metrics and metrics only.
- Alarms can be set to take action based on metrics data.



CloudWatch Alarm

Alarm

Alarm Threshold and Period. (Threshold of 75% for 3 consecutive times)



Alarm States

- OK – Within Threshold.
- ALARM – Crossed Threshold.
- INSUFFICIENT_DATA – Metric not available/ Missing data (Good, Bad, Ignore, Missing).

Demo 2: Alarm

Demo 2: Alarm

Alarm

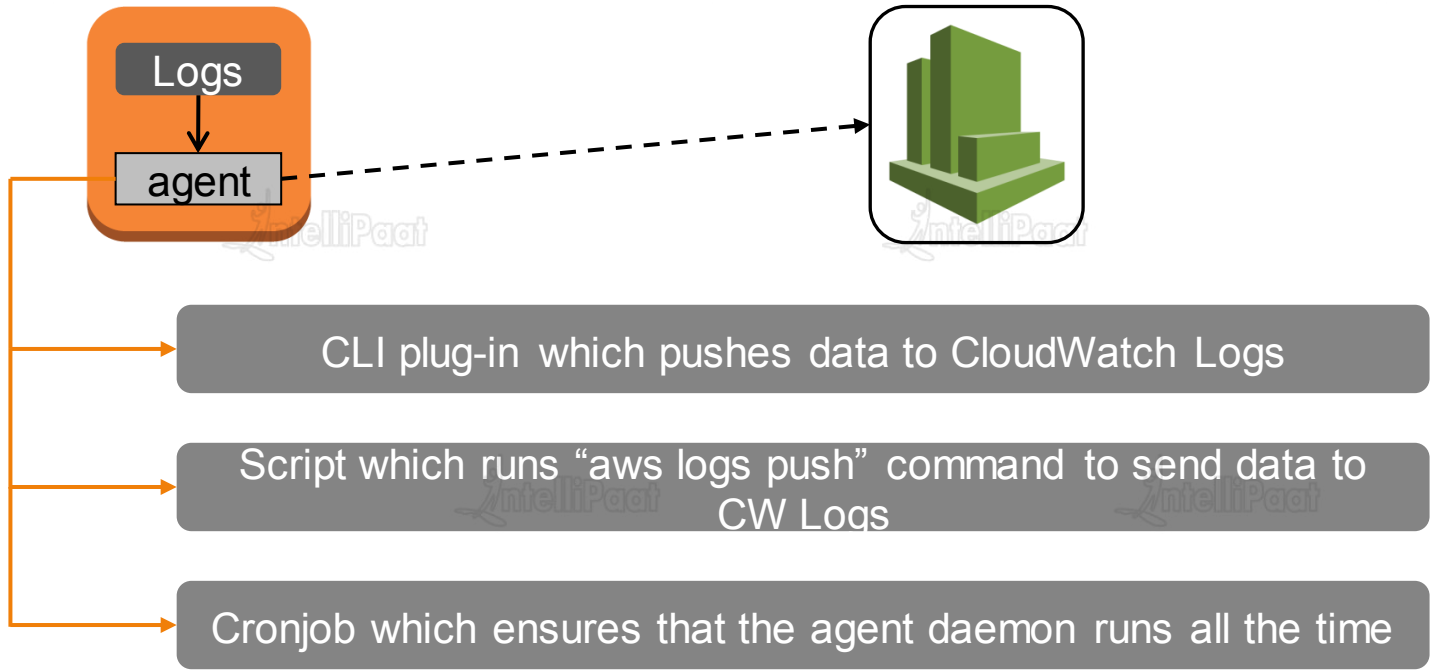
- Create an Alarm.
- Alarm should monitor CPU Utilization of an EC2 instance.
- Decide on the Threshold and period.
- Provision all the 3 actions – SNS, Auto Scale and EC2 instance termination.
- Create Billing Alarm.

CloudWatch Logs

CloudWatch Logs

Logs

- CloudWatch logs are used to monitor, store and access log files from various AWS resources including EC2 etc.
- How does it work:



CloudWatch Logs

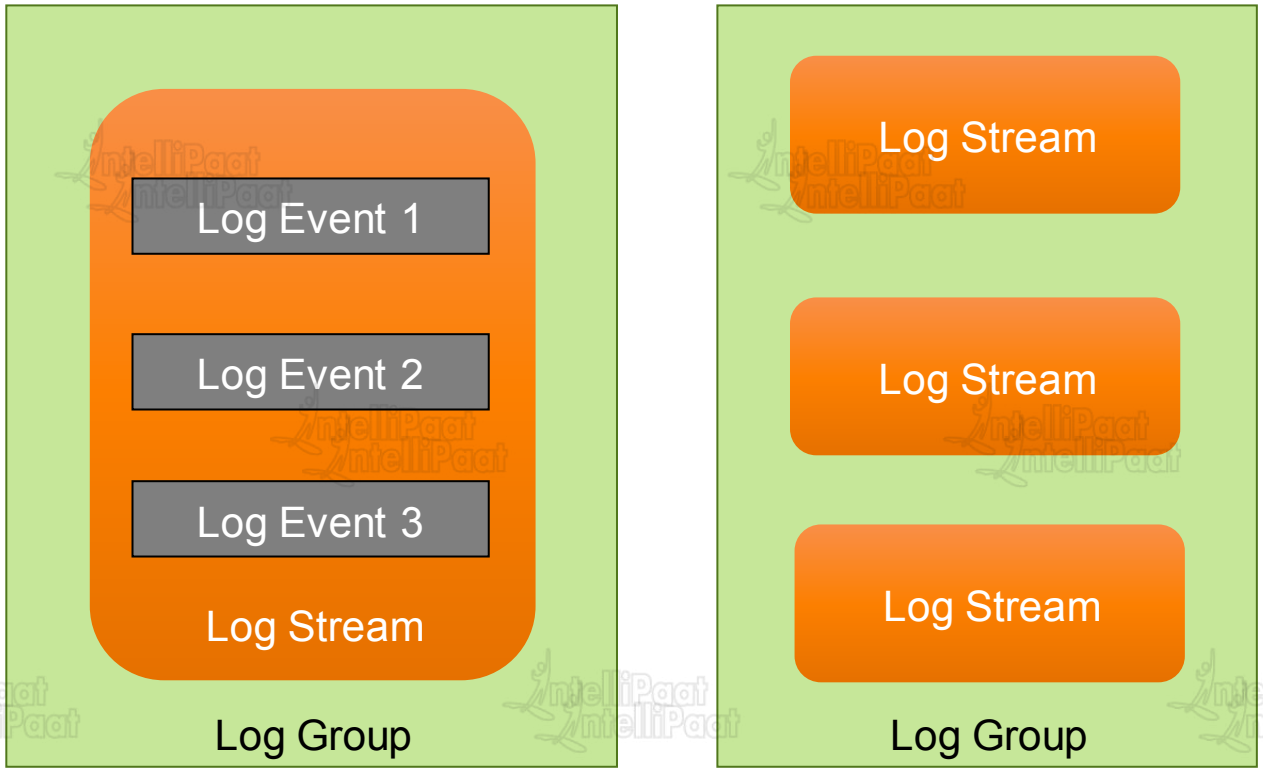
CloudWatch Log Components

Log Events: Record of some activity recorded by the application being monitored.

Log Streams: Sequence of log events from the same source

Log Groups: Group of Log Streams.

Metric Filters: Customized metrics created from received log data.



CloudWatch Logs

Installing Logs Agent

- Install and configure the agent

```
sudo yum install -y awslogs
```

```
/etc/awslogs/awscli.conf
```

```
/etc/awslogs/awslogs.conf
```

```
sudo service start awslogs
```

```
/var/log/awslogs.log
```

```
sudo chkconfig awslogs on
```

CloudWatch Logs

Log Config File

- Config File: Contains information needed by “aws logs push” command.

General Section:
state_file
logging_config_file

Logstream Section:
log_group_name = value
log_stream_name = value
file = value
batch_count = integer
batch_size = integer

Demo 3: CloudWatch Logs

Demo 3: CloudWatch Logs

- Configure log agent in an EC2 instance.
- Create log group.
- Use Filter patterns.
- Export data to S3.

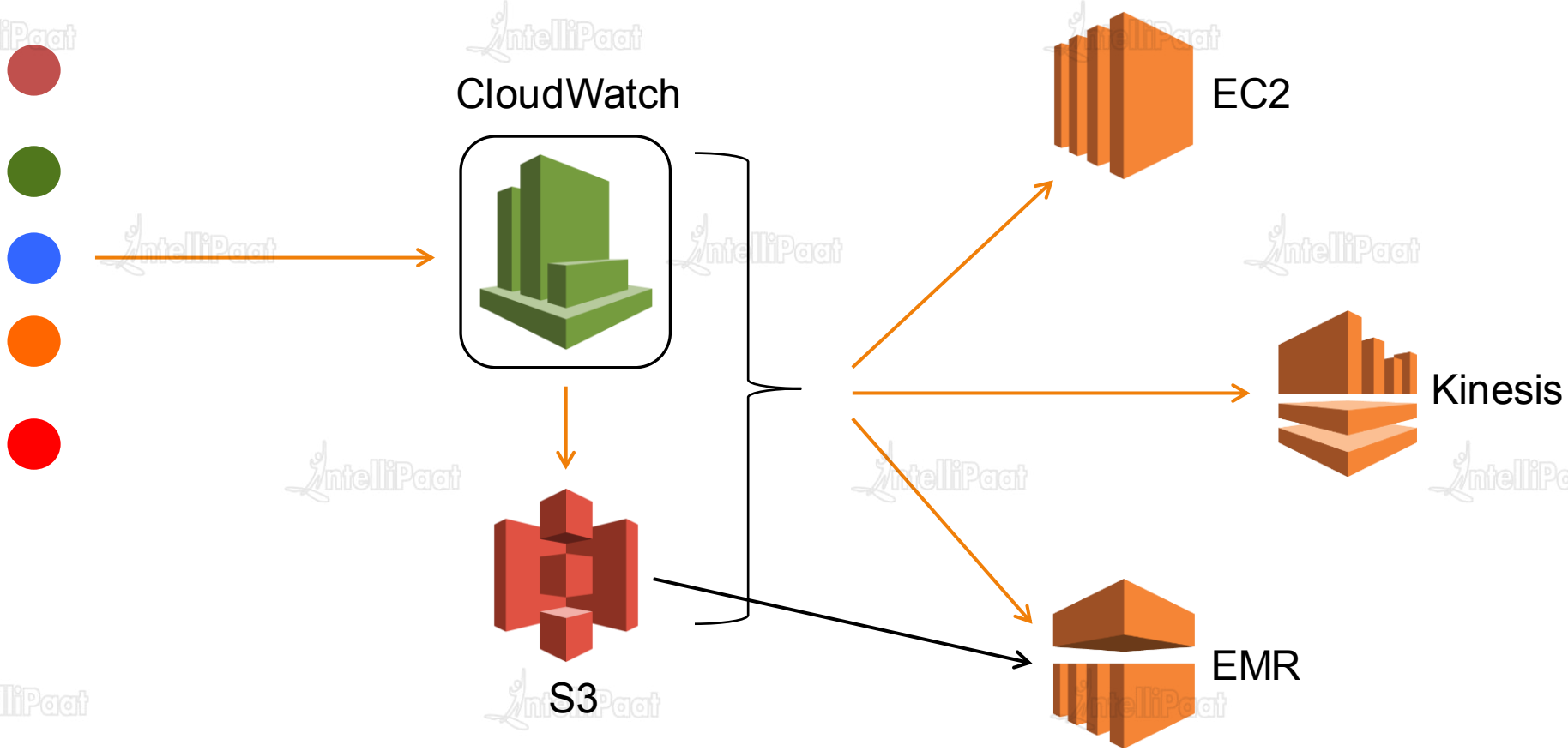
Pricing

CloudWatch Pricing (us-east-1)

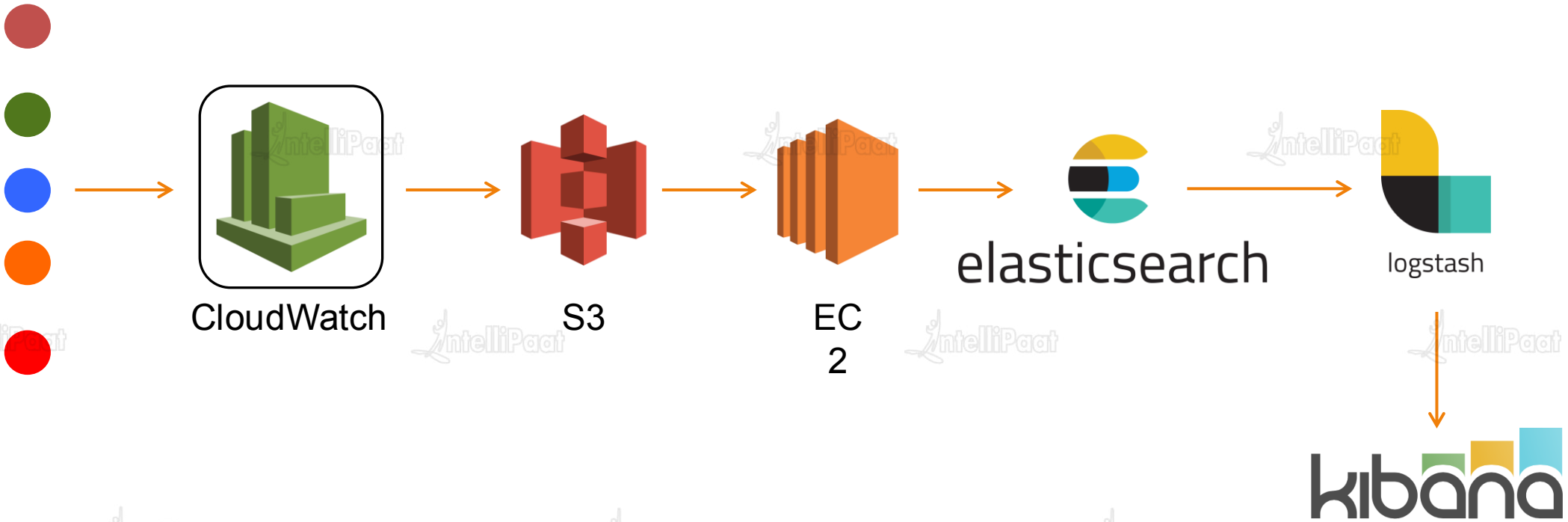
- Free Tier
  - 3 dashboards up to 50 metrics per month
  - Basic monitoring at 5 mins interval of EC2, EBS, ELB, RDS are free.
- https://aws.amazon.com/cloudwatch/pricing/
- Pricing
  - Dashboards: $3.00 per dashboard per month
  - Detailed monitoring for EC2 instances
  - Custom Metrics
  - Alarms: $0.10 per alarm/month
  - CloudWatch Logs
  - CloudWatch Events

Design Patterns

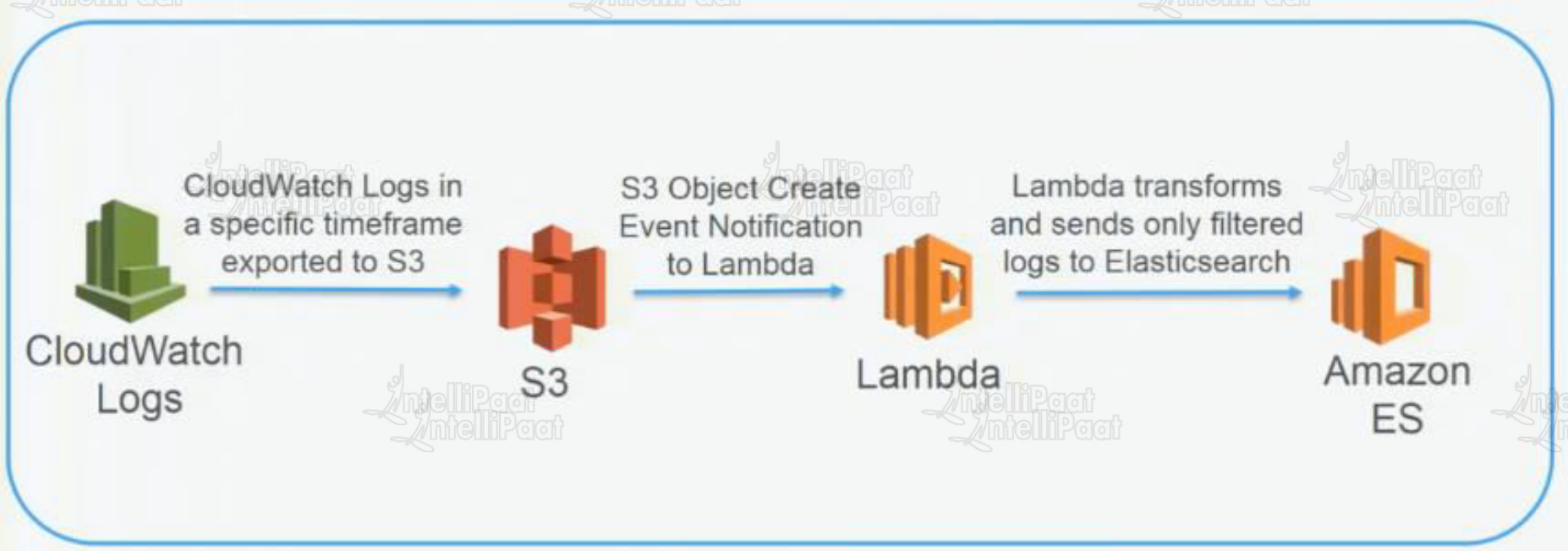
Design Patterns



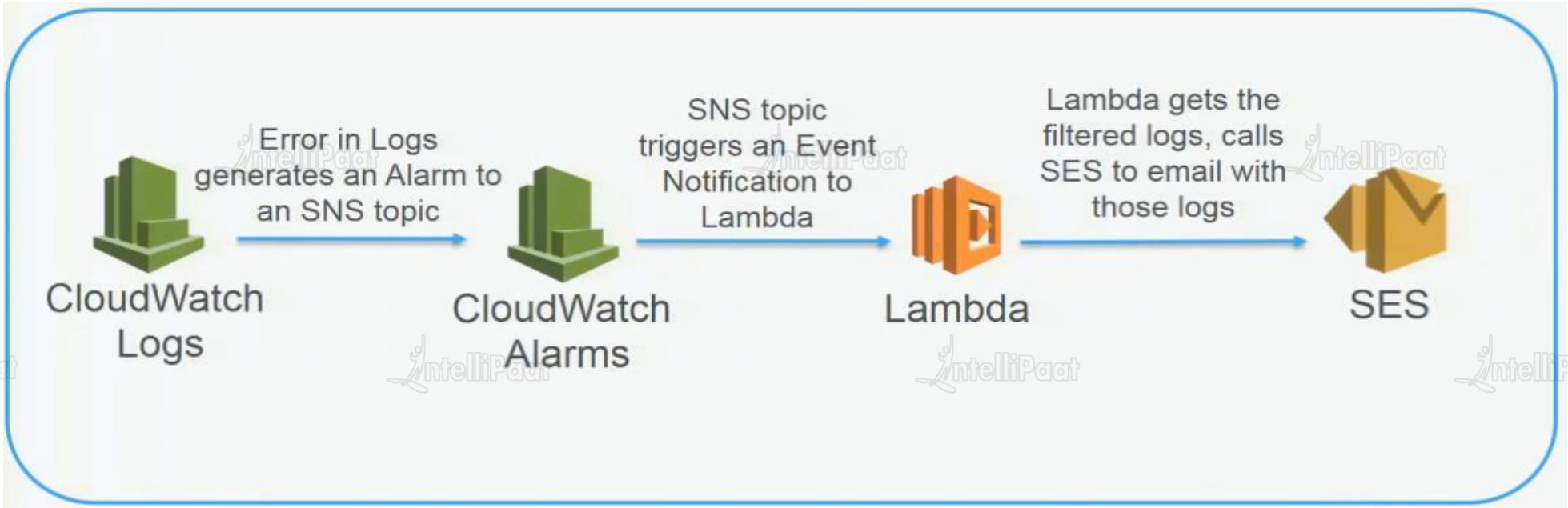
Design Patterns



Design Patterns



Design Patterns



Design Patterns

Quiz

Quiz

1. When CloudWatch state is ALARM, does it mean that the metric is within the defined threshold?

A. Yes

B. No

Quiz

2. Is CloudWatch DisReadOps Metric used to calculate the average I/O operations per second (IOPS)

A. Yes

B. No

Quiz

3. Cloudwatch retains metric data points with a period of 60 seconds for 15 days

A. Yes

B. No

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24X7 Chat with our Course Advisor