UC Job Builder on Amazon Web Services Cloud

UCOP

PRESENTED BY

BRET DOWELL

RAJANI PRAKASH

SUSHANT PRASAD

UNIVERSITY
OF
CALIFORNIA

8/14/2018

UC Job Builder

- The UC Job Builder is an application for using Job Standards to create individual position descriptions.

- A Job Standard is a pre-defined template to describe the Scope, Key Responsibilities, and Knowledge and Skills requirements of a specific job level within a job family.

- The Job Standards are categorized by the following:
  - Job Family
  - Job Function
  - Job Category
  - Job Code

UC Job Builder

- The compensation team at UCOP maintains the job templates for each title code that is in use within the UC system.

- Campuses can use those templates to generate job descriptions for their staff or refer to job descriptions from other campuses.

- Currently, Job Builder is being used by UC Office of the President, UC Berkeley, UC Merced and UC ANR.

UC Job Builder

- The Job Builder application was an on premise application originally hosted by UC Merced.
- We at UCOP TDS-EAS Team (Donna Yamasaki's team) re-wrote the application and hosted it on the Amazon Web Services (AWS) cloud for use by UC system wide.
- This is the first transactional web application by UCOP hosted on the AWS cloud.
- We would like to share our journey, learnings, experiences and the challenges faced during the AWS cloud implementation.

Technology Stack Version 3.0

Originally

- Technology Stack
  - ASP.NET Webforms
  - JavaScript
  - C#
  - ADO.NET/Entity Framework
- Tooling
  - Visual Studio
  - CodeSmith
  - NetTiers
- SSO
  - LDAP, CAS, Shibboleth (depending on location)

Technology Stack Version 4.0

Re-Written by UCOP

- Technology Stack
  - ASP.NET MVC
  - JavaScript
  - C#
  - Entity Framework
  - Bootstrap
  - SQL Server, SSIS
- Tooling
  - Visual Studio
  - TFS
- SSO
  - Shibboleth

jobbuilder.ucmerced.edu

jobbuilder.ucop.edu

UNIVERSITY OF CALIFORNIA
Agriculture and Natural Resources
We are a network of scientists connecting the people of California with practical, non-biased UC research.
We are problem-solvers, catalysts, collaborators, educators and stewards of the land, living in the communities we serve.
#WeAreUCANR
jobbuilder.ucanr.edu

Jobbuilder Berkeley

Job Builder (Before)

Old Technology, discrete applications, no standardization of Job Standards across campuses. Deployed per campus, localized branding.

ucjobbuilder.ucop.edu

UNIVERSITY OF CALIFORNIA
Agriculture and Natural Resources
We are a network of scientists connecting the people of California with practical, non-biased UC research.
We are problem-solvers, catalysts, collaborators, educators and stewards of the land, living in the communities we serve.
#WeAreUCANR

UC Job Builder (New)

Common URL, Shibboleth login, Job Standards Standardization driven from UCOP, Newer Technologies; Search/View Job Standards and create Position Descriptions

UNIVERSITY
OF
CALIFORNIA

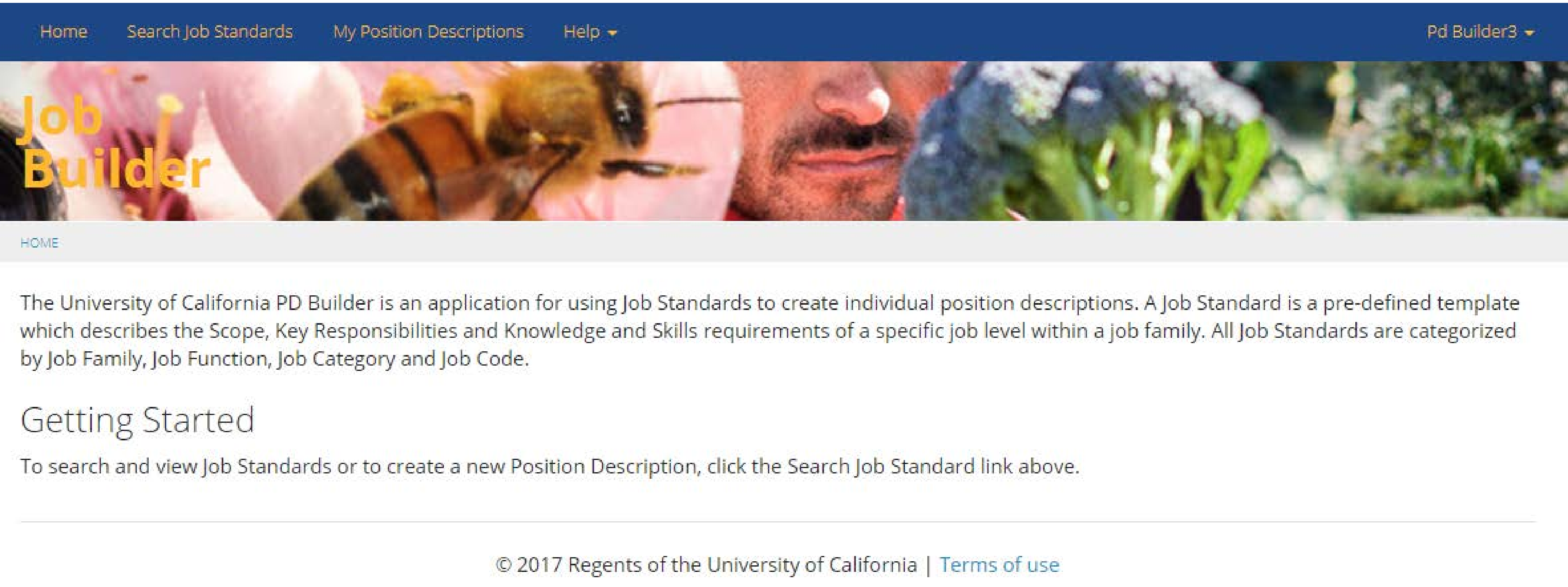
8/14/2018

UC Job Builder

Localized Branding and look and feel per Campus:

- **campus specific graphics**
- **name**
- **data**

UC Job Builder Home Page (ANR)



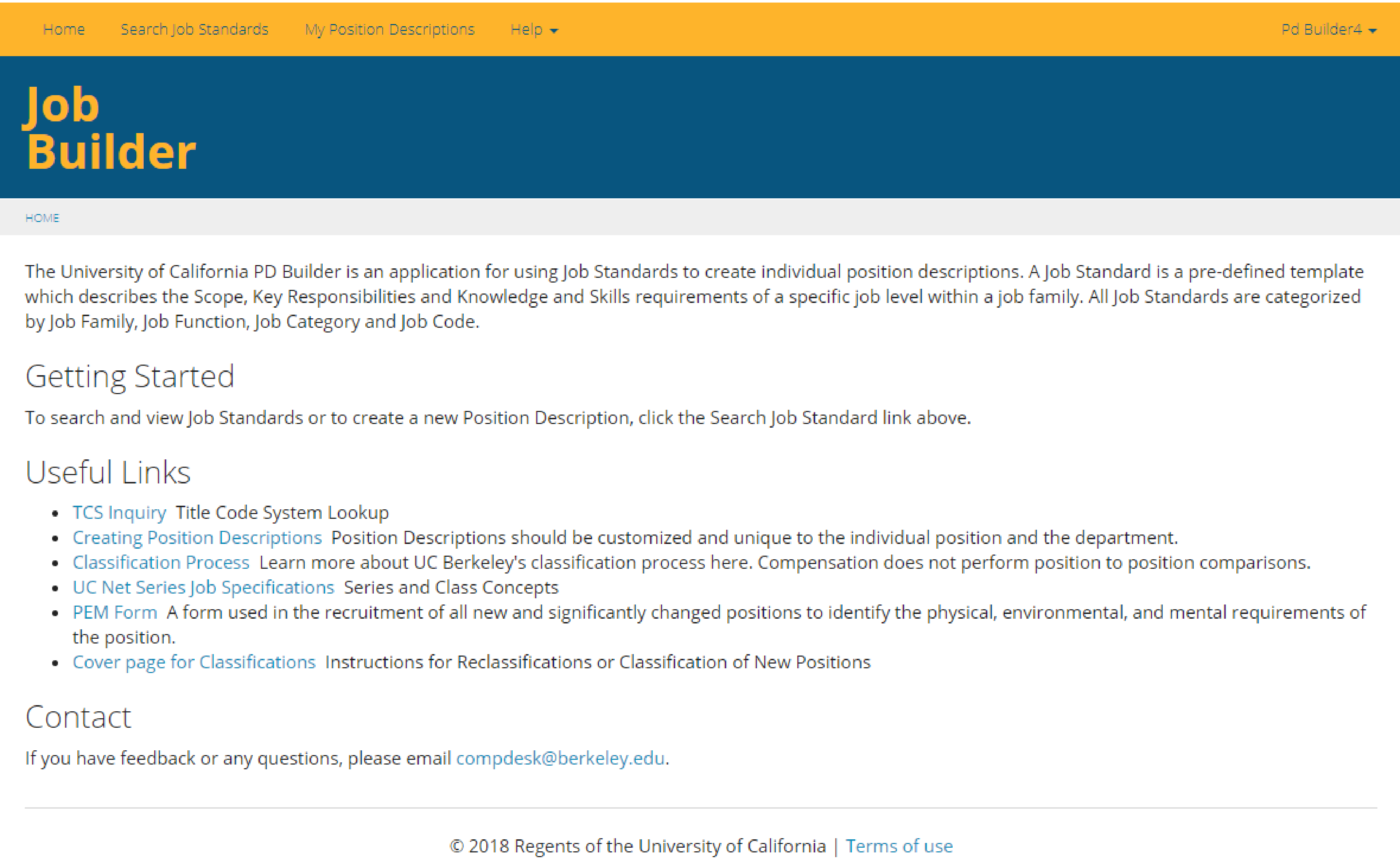
The University of California PD Builder is an application for using Job Standards to create individual position descriptions. A Job Standard is a pre-defined template which describes the Scope, Key Responsibilities and Knowledge and Skills requirements of a specific job level within a job family. All Job Standards are categorized by Job Family, Job Function, Job Category and Job Code.

Getting Started

To search and view Job Standards or to create a new Position Description, click the Search Job Standard link above.

© 2017 Regents of the University of California | Terms of use

UC Job Builder Home Page (Berkeley)



The University of California PD Builder is an application for using Job Standards to create individual position descriptions. A Job Standard is a pre-defined template which describes the Scope, Key Responsibilities and Knowledge and Skills requirements of a specific job level within a job family. All Job Standards are categorized by Job Family, Job Function, Job Category and Job Code.

Getting Started

To search and view Job Standards or to create a new Position Description, click the Search Job Standard link above.

Useful Links

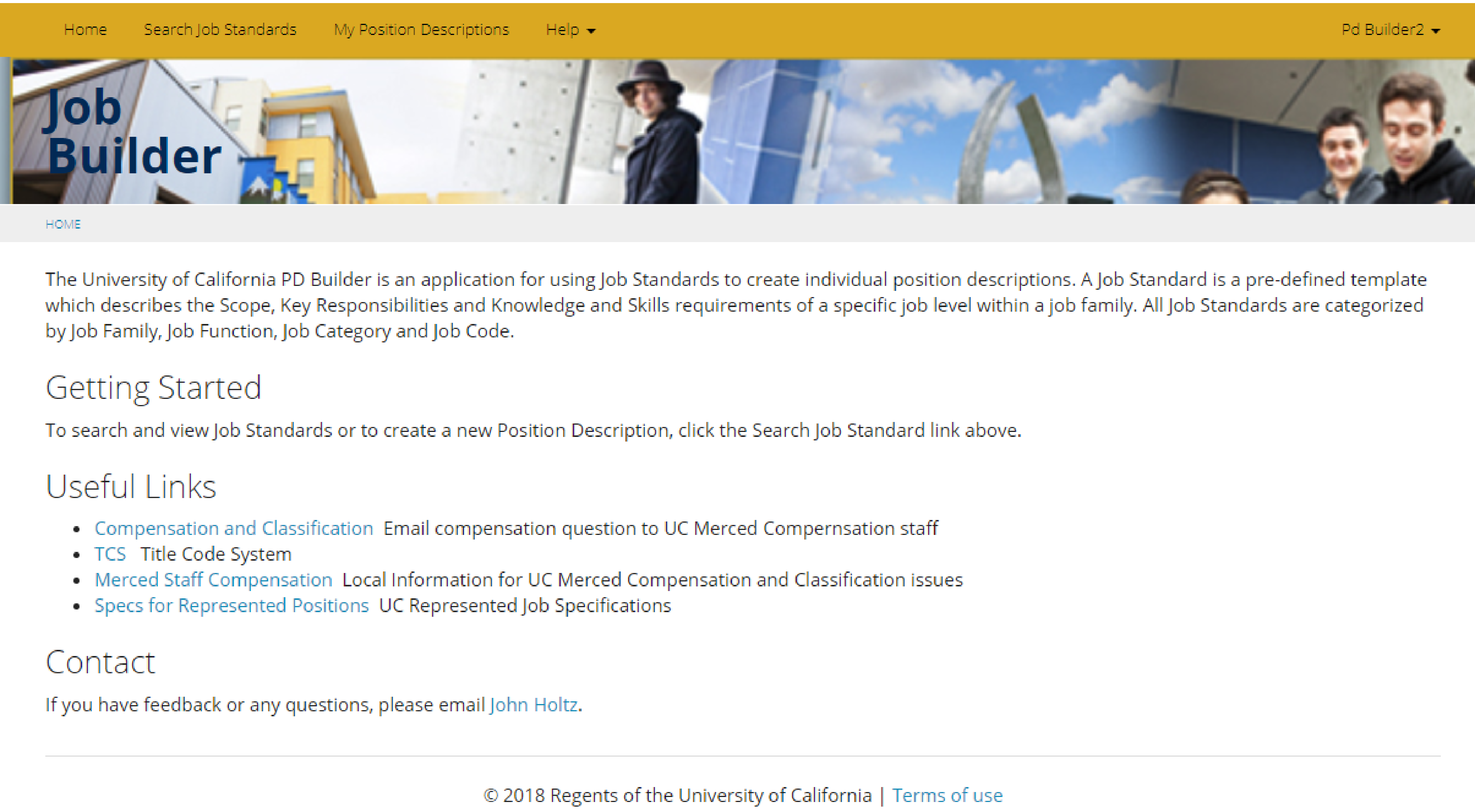
- TCS Inquiry Title Code System Lookup
- Creating Position Descriptions Position Descriptions should be customized and unique to the individual position and the department.
- Classification Process Learn more about UC Berkeley's classification process here. Compensation does not perform position to position comparisons.
- UC Net Series Job Specifications Series and Class Concepts
- PEM Form A form used in the recruitment of all new and significantly changed positions to identify the physical, environmental, and mental requirements of the position.
- Cover page for Classifications Instructions for Reclassifications or Classification of New Positions

Contact

If you have feedback or any questions, please email compdesk@berkeley.edu.

© 2018 Regents of the University of California | Terms of use

UC Job Builder Home Page (Merced)



The University of California PD Builder is an application for using Job Standards to create individual position descriptions. A Job Standard is a pre-defined template which describes the Scope, Key Responsibilities and Knowledge and Skills requirements of a specific job level within a job family. All Job Standards are categorized by Job Family, Job Function, Job Category and Job Code.

Getting Started

To search and view Job Standards or to create a new Position Description, click the Search Job Standard link above.

Useful Links

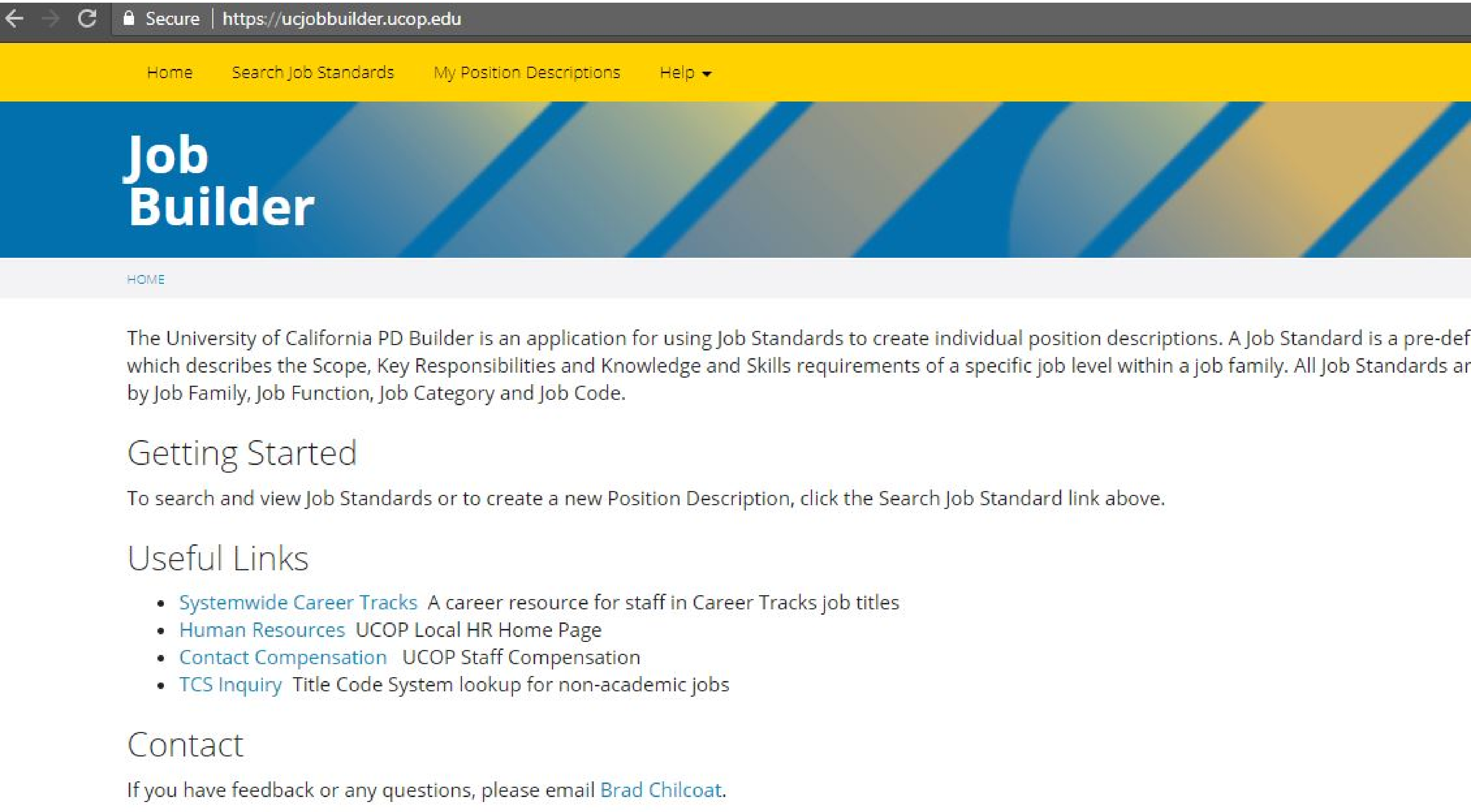
- Compensation and Classification Email compensation question to UC Merced Compernsation staff
- TCS Title Code System
- Merced Staff Compensation Local Information for UC Merced Compensation and Classification issues
- Specs for Represented Positions UC Represented Job Specifications

Contact

If you have feedback or any questions, please email John Holtz.

© 2018 Regents of the University of California | Terms of use

UC Job Builder Home Page (UCOP)



The University of California PD Builder is an application for using Job Standards to create individual position descriptions. A Job Standard is a pre-def
which describes the Scope, Key Responsibilities and Knowledge and Skills requirements of a specific job level within a job family. All Job Standards ar
by Job Family, Job Function, Job Category and Job Code.

Getting Started

To search and view Job Standards or to create a new Position Description, click the Search Job Standard link above.

Useful Links

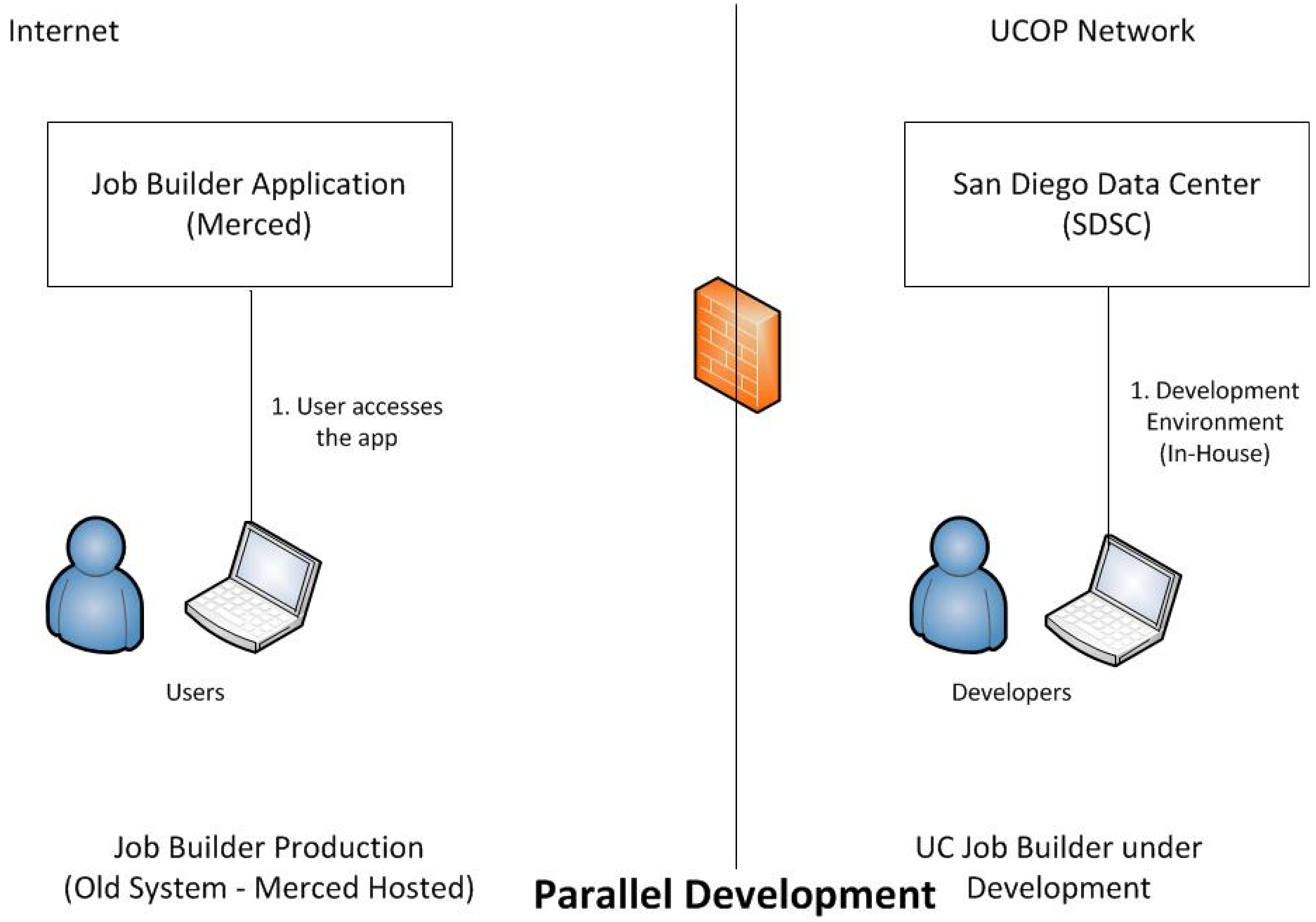
- Systemwide Career Tracks A career resource for staff in Career Tracks job titles
- Human Resources UCOP Local HR Home Page
- Contact Compensation UCOP Staff Compensation
- TCS Inquiry Title Code System lookup for non-academic jobs

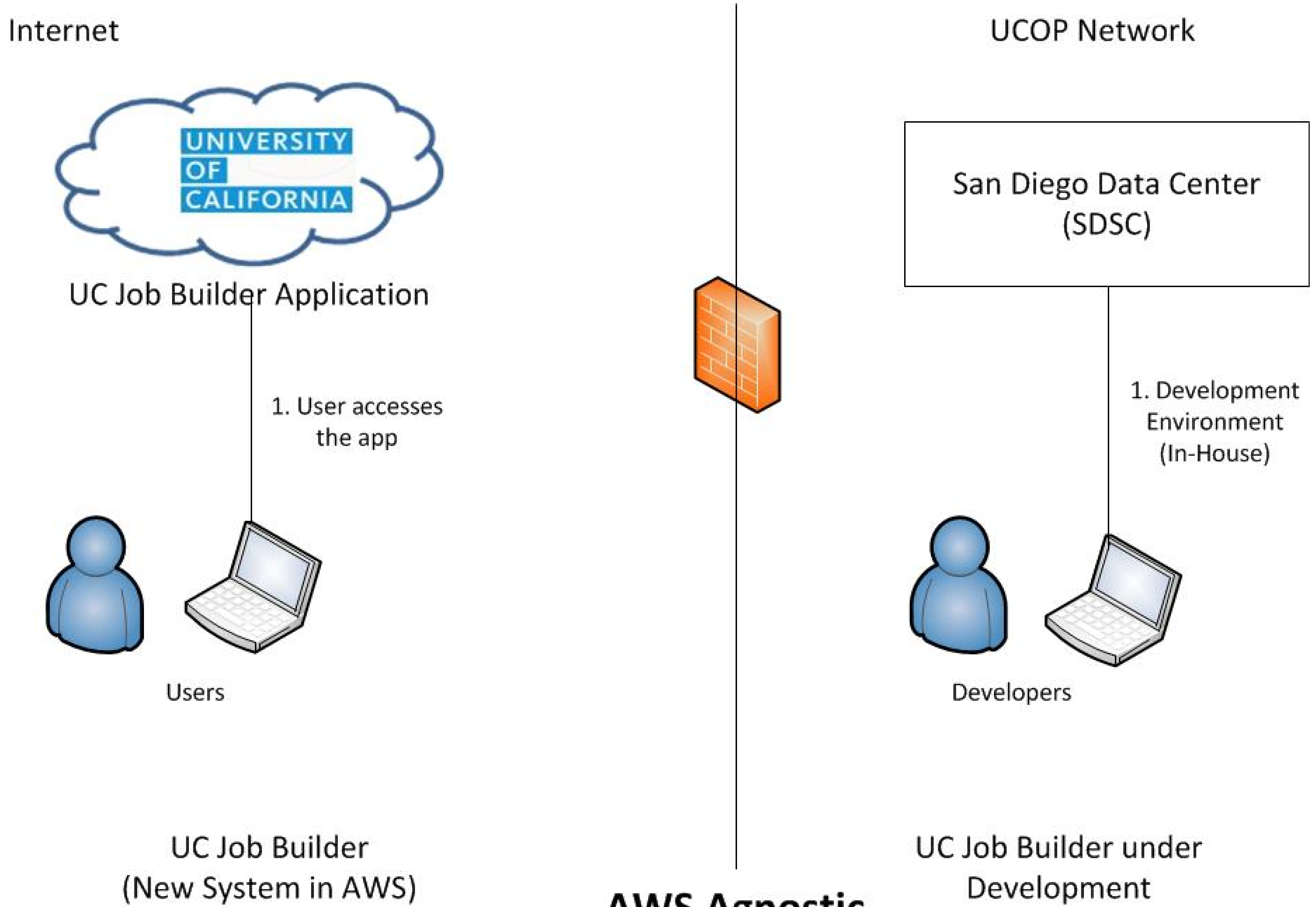
Contact

If you have feedback or any questions, please email Brad Chilcoat.

Progress is Better than Perfection

Our Journey to the AWS cloud…


Internet
UCOP Network
Job Builder Application (Merced)
San Diego Data Center (SDSC)
1. User accesses the app
1. Development Environment (In-House)
Users
Developers
Job Builder Production (Old System - Merced Hosted)
UC Job Builder under Development
Parallel Development


Internet
UCOP Network
UNIVERSITY OF CALIFORNIA
UC Job Builder Application
San Diego Data Center (SDSC)
1. User accesses the app
1. Development Environment (In-House)
Users
Developers
UC Job Builder (New System in AWS)
UC Job Builder under Development

AWS Agnostic Development

Why Cloud ?

- UCOP TDS (Technology Delivery Services) Cloud first strategy
- Enables business to be more agile
- More responsive to changing customer needs
- Time to market is drastically reduced
- Leverage the economies of scale
- Frees up the infrastructure teams
    - to design and innovate
    - contribute to change

Why Amazon Web Services ?

- On-Demand Cloud Computing platforms
- Best in class cloud infrastructure
- All Service Offerings
  - SaaS (Software as a Service)
  - PaaS (Platform as a Service)
  - IaaS   (Infrastructure as a Service)
- Pace of Innovation
- Pay for use Pricing
- Many other notable features

On-premise Traditional Data Center

- Engineering controls every facet of:
  - network
  - storage
  - servers
- e.g.
  - memory
  - brand
  - pre-configured sizing and everything.

System Requirements

- Fine grain control of the environment – EC2 and the various server/network components
- Environment should be scripted - so setup and tear down is easy
- Comply with the network and security needs – proper controls
- Application Alerting and EOC integration
- Secure and Controlled access to the environment
- Security Information and Event Management (SIEM) Integration

Initial Experiments with

- Elastic Beanstalk
- Docker

AWS Elastic Beanstalk

- Service for deploying Web applications
- Setting up the environment stack

Pros :

- deployment
- capacity provisioning
- load balancing
- auto-scaling
- application health monitoring

Cons :

- Team losing fine gain control over the environment
- Toolkit did not work for us

Result: Abandon the approach

Initial Experiments: Docker

- Supports OS-level Virtualization/Containerization
  - Pros:
    - Extremely lightweight and modular
    - Flexible containers – easy to move
  - Cons:
    - Very unstable on the windows 2012 server environment at that time (2017 fall)

Result: Abandon the approach

AWS Key Design Questions

- Infrastructure
  - Connectivity with on premise infrastructure to run business processes
  - VPN connection between onpremise and AWS needed
  - Server OS and version for Database and Application servers
  - Application, database, third party tools, script and batch components
  - Is a DR site needed
  - Application load balancer needed
- Database
  - Type of Data- PII, PCI, HIPPA, etc.
  - Read replica for adhoc queries or reporting needed
  - How many databases, database size
  - Database platform and version and size
  - What is the data type? (OLTP/DWH)
  - How much Downtime allowed
  - Expected database growth rate by database

AWS Key Design Questions

- Application
  - Transition time allowed to the Failover site
  - Application usage pattern
  - How is the application accessed - internet or intranet
  - Application authentication scheme
  - Application logging to flat files or to database tables
  - Operational dashboard for key metrics and logging information
  - Expected user base growth by application
- Security & Compliance
  - Authentication using Active Directory, LDAP, Federated AD,Etc.
  - Compliance requirements such as PCI, HIPPA, etc.
  - Database encryption
- Migration
  - Allowed downtime
  - Cut-over planning, data migration etc.

AWS Key Design Questions

- Performance/Load
  - Total user connections, Peak user connections, Requests/sec
  - Peak and average CPU and Memory utilization
  - Database IOPS, average, Peak.
  - Application IOPS, average , peak
  - Performance SLA's such as Response times, transactional performance SLAs
- Scripts
  - Scripting languages in use
  - SQL version dependency etc.
- Scheduled Jobs
  - SSIS Jobs needed
  - Database connectivity needed
  - ETL or Data import/Export processes

AWS Key Design Questions

- Reporting
  - Reporting systems - SSRS/SSAS etc
  - Read-only databases for reporting
  - Access to reporting

- Testing
  - Test scripts
  - Resources for UAT
  - Performance benchmarks
  - Functional testing

AWS Key Design Elements

Relational Database

Amazon RDS vs Running SQL Server on EC2 ?

Amazon RDS

Amazon Relational Database Service (Winner for us !)

Easy to set up, operate, and scale

- **Automated –**
  - **Installation, disk provisioning & management**
  - **minor version upgrades, patching, failed instance replacement, database backup and recovery**
- **Automated**
  - **Multi-AZ (Availability Zone) synchronous replication**
- **Fully managed by AWS - highly available and scalable environment**
- **Amazon handles the day-to-day database management**
- **Frees us to focus on higher-level tasks - schema optimization, query tuning, and application development**
- **High availability and automated failover capabilities**

UC Job Builder RDS Setup

- The SQL server instance used by Job builder application is set up as a Multi-AZ setup.

- The primary server is located in one availability zone in us-west-2 region and a secondary server is located in another availability zone within the same region.

- SQL server mirroring: Primary node to the secondary node. Cross-regional mirroring is not available currently.

- Automatic Failover: Primary RDS instance fails over to the secondary instance in another AZ in case of any irrecoverable issues with the primary instance. SQL server endpoint remains unchanged and no connection string updates need to be performed for pointing any dependent applications to the failed-over instance

- Backups: RDS provides a full back up once a day and transaction log backups every 5 minutes. Full backups are done using snapshots. RDS provides Point of Restore option

SQL Server Integration Services (SSIS)

- SSIS Packages and Jobs
- To import the data from various campuses.

The Challenge:

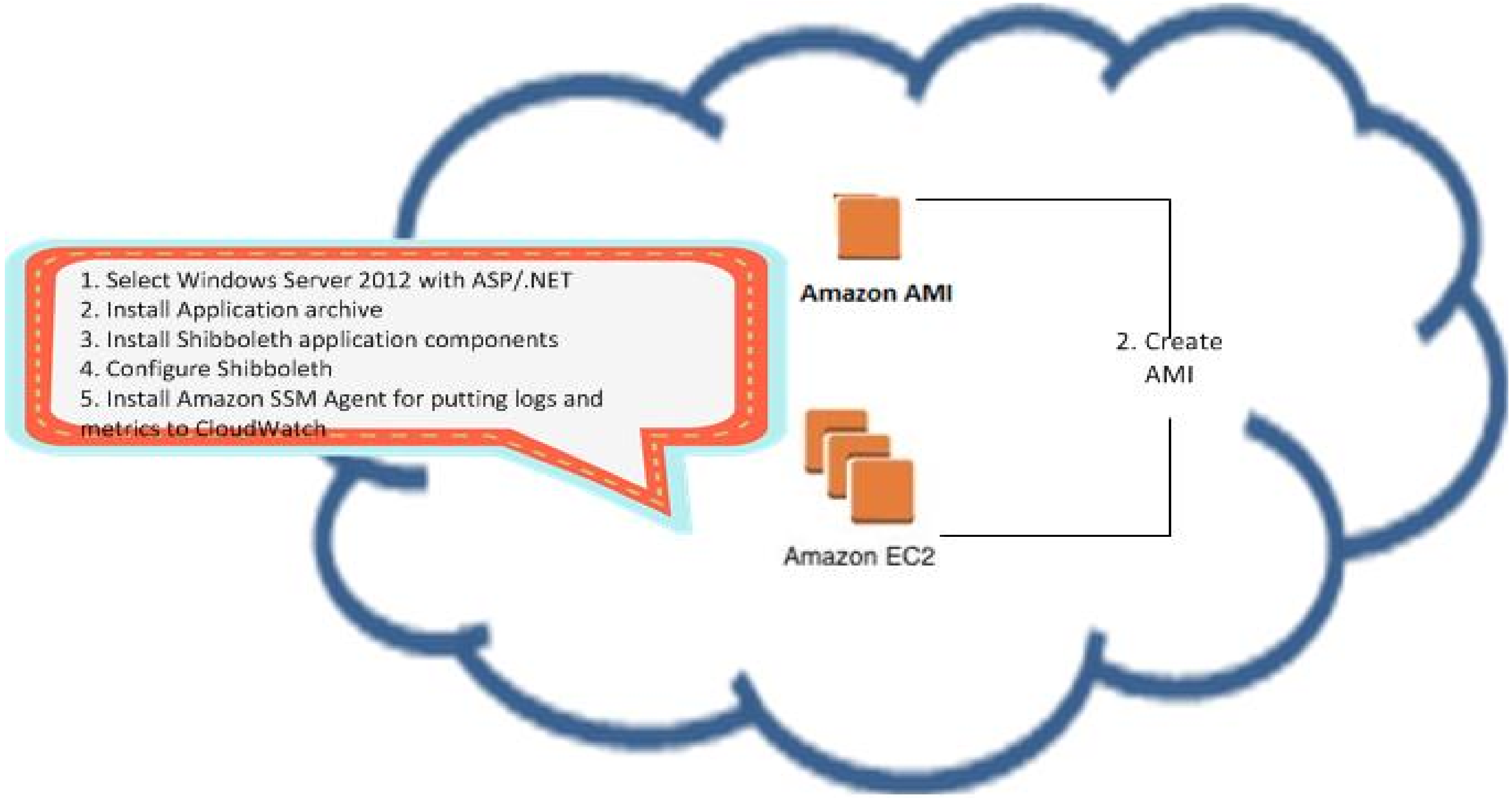
- RDS does not support SSIS i.e. primarily just the database component

- We had to run the SSIS job server independently

- Also, connectivity to campuses to retrieve the data

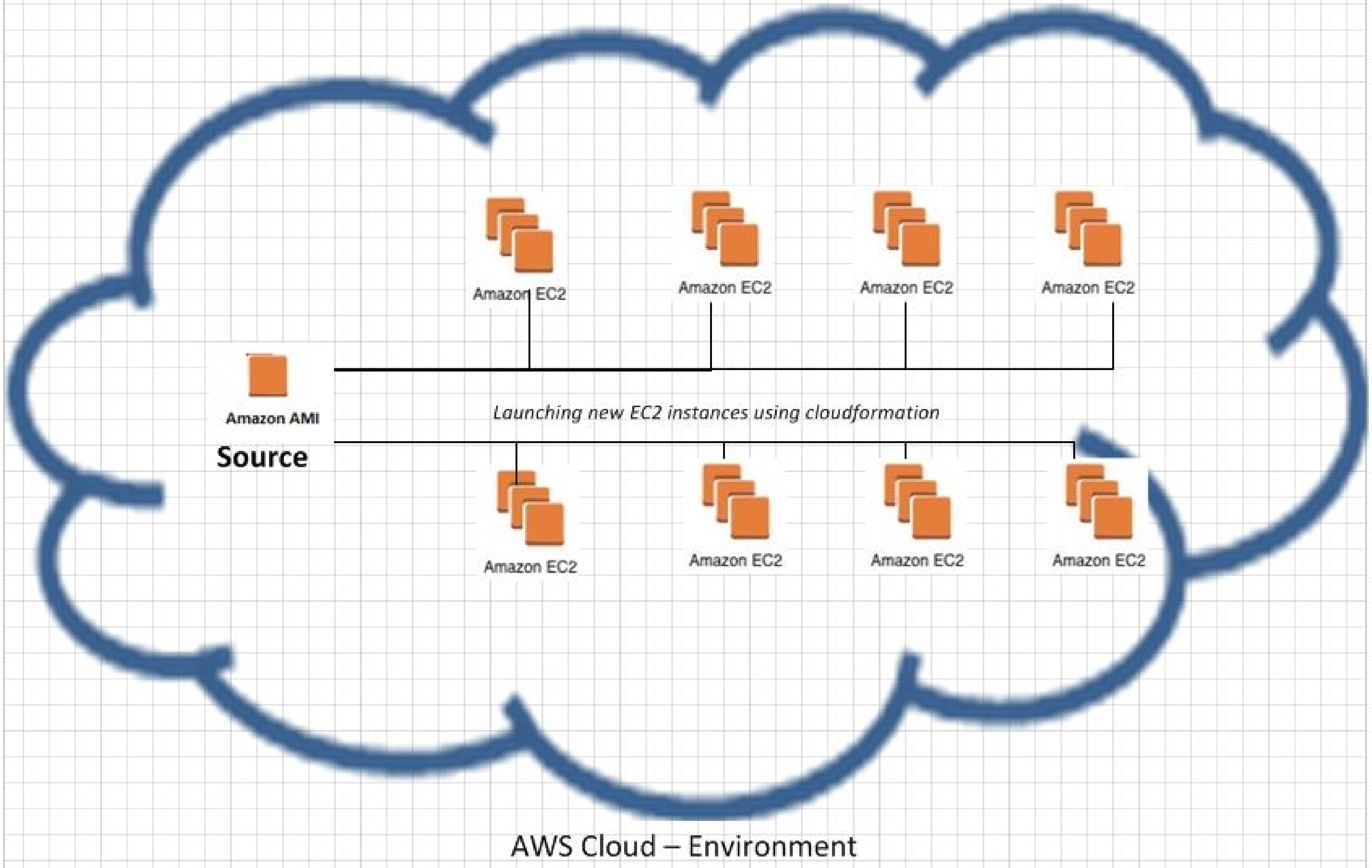
Web Server Setup and AMI Creation

- Windows Server 2012 with ASP/.NET
- Create an Amazon Machine Image to launch an instance
- AMI Provides information to launch an instance (virtual Server in the cloud)
- AMI can Launch multiple instances with same configuration
- The AMI Includes
    - A template for the root volume for the instance for e.g. OS, App Server, applications etc.
    - Launch permissions to control which AWS accounts can use the AMI to launch instances
    - A block device mapping i.e. the volumes to attach to the instance when it's launched


1. Select Windows Server 2012 with ASP/.NET
2. Install Application archive
3. Install Shibboleth application components
4. Configure Shibboleth
5. Install Amazon SSM Agent for putting logs and metrics to CloudWatch
Amazon AMI
2. Create AMI
Amazon EC2

AWS Cloud Environment

JobBuilder Application Creating First AMI



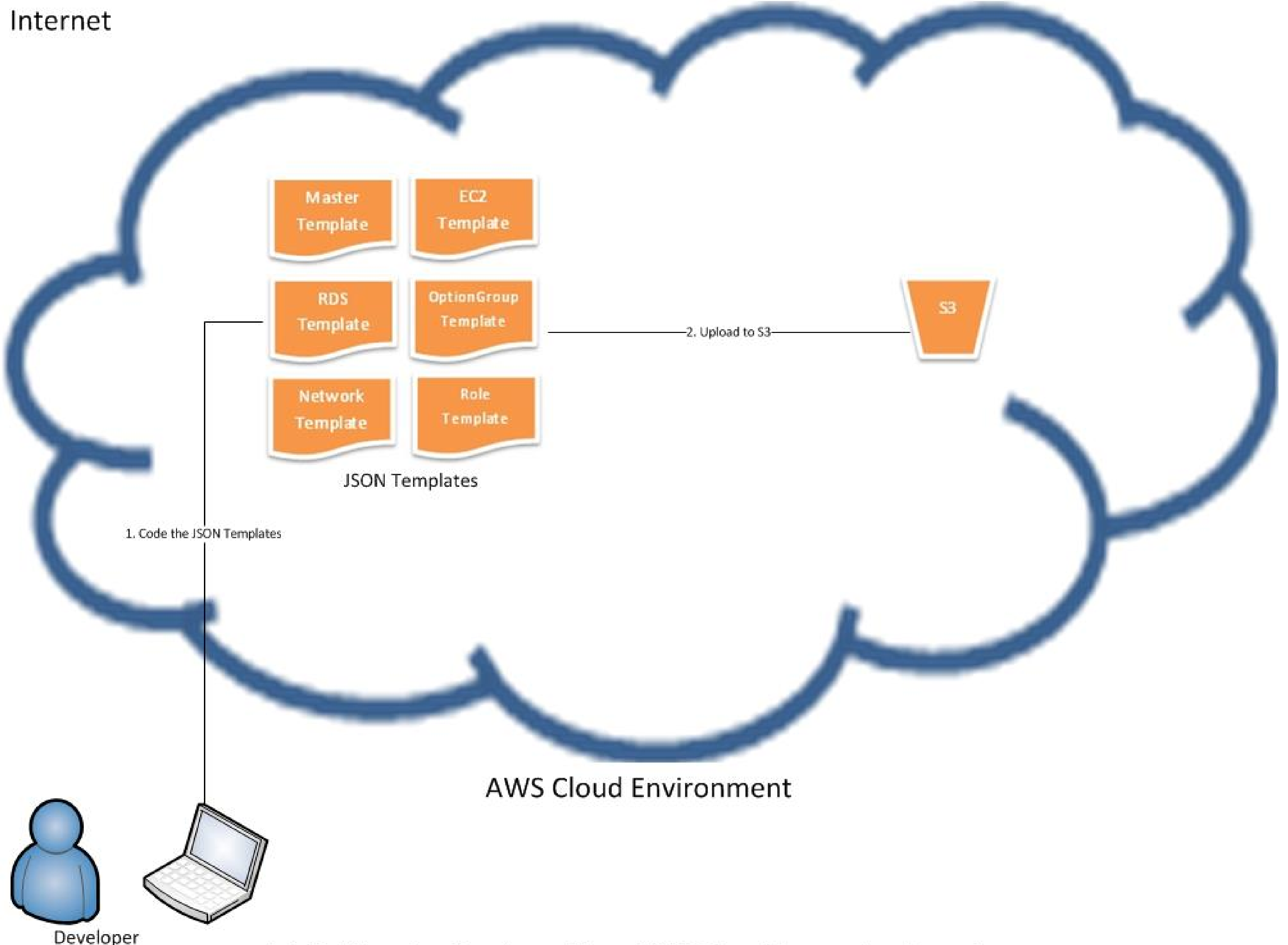
JobBuilder Application Creating New Instances from AMI

Infrastructure as Code

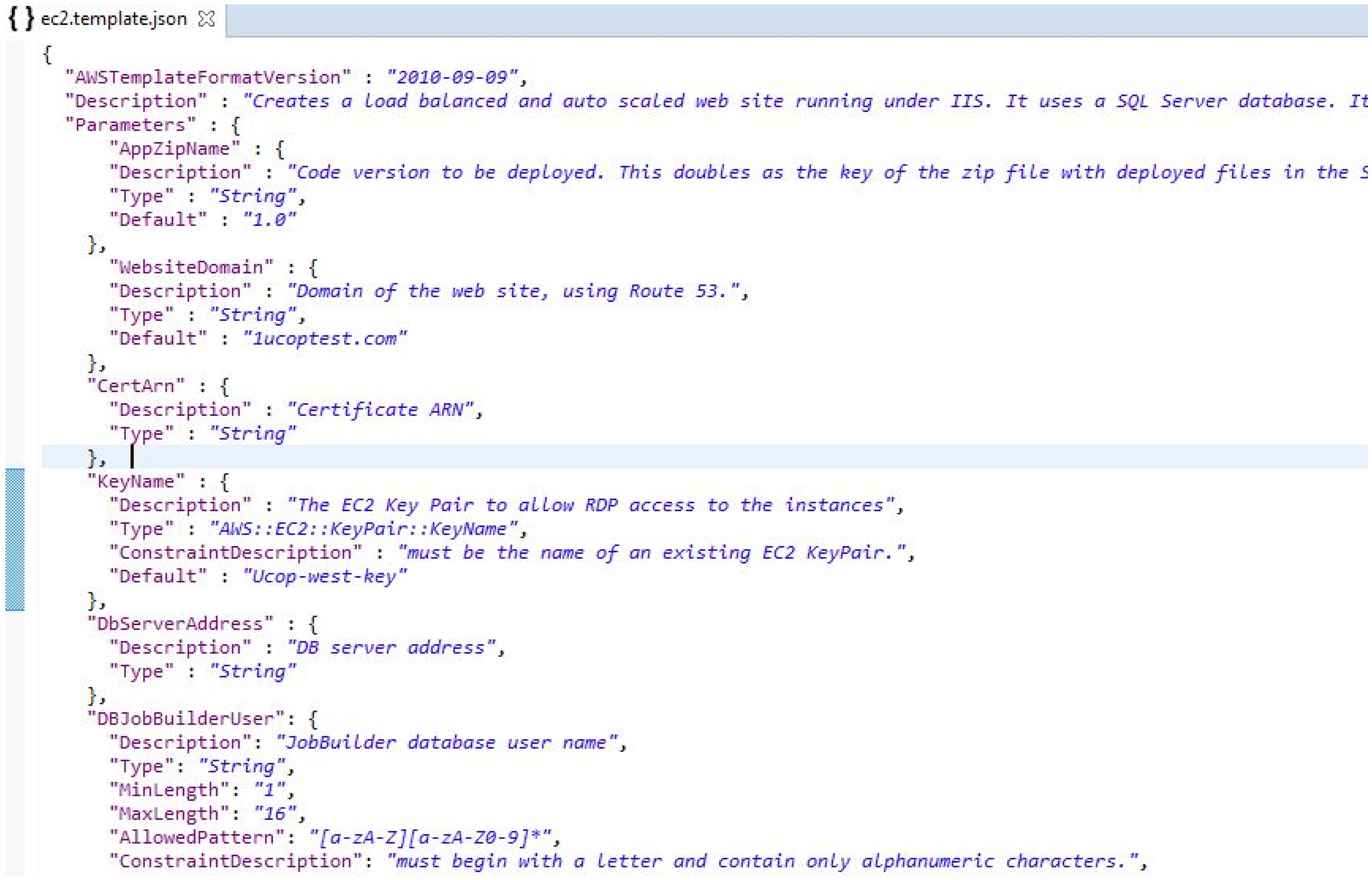
Using CloudFormation

- A common language to describe and provision all the infrastructure resources in the cloud environment.

- Templates to Describe AWS Resources & Associated Dependencies

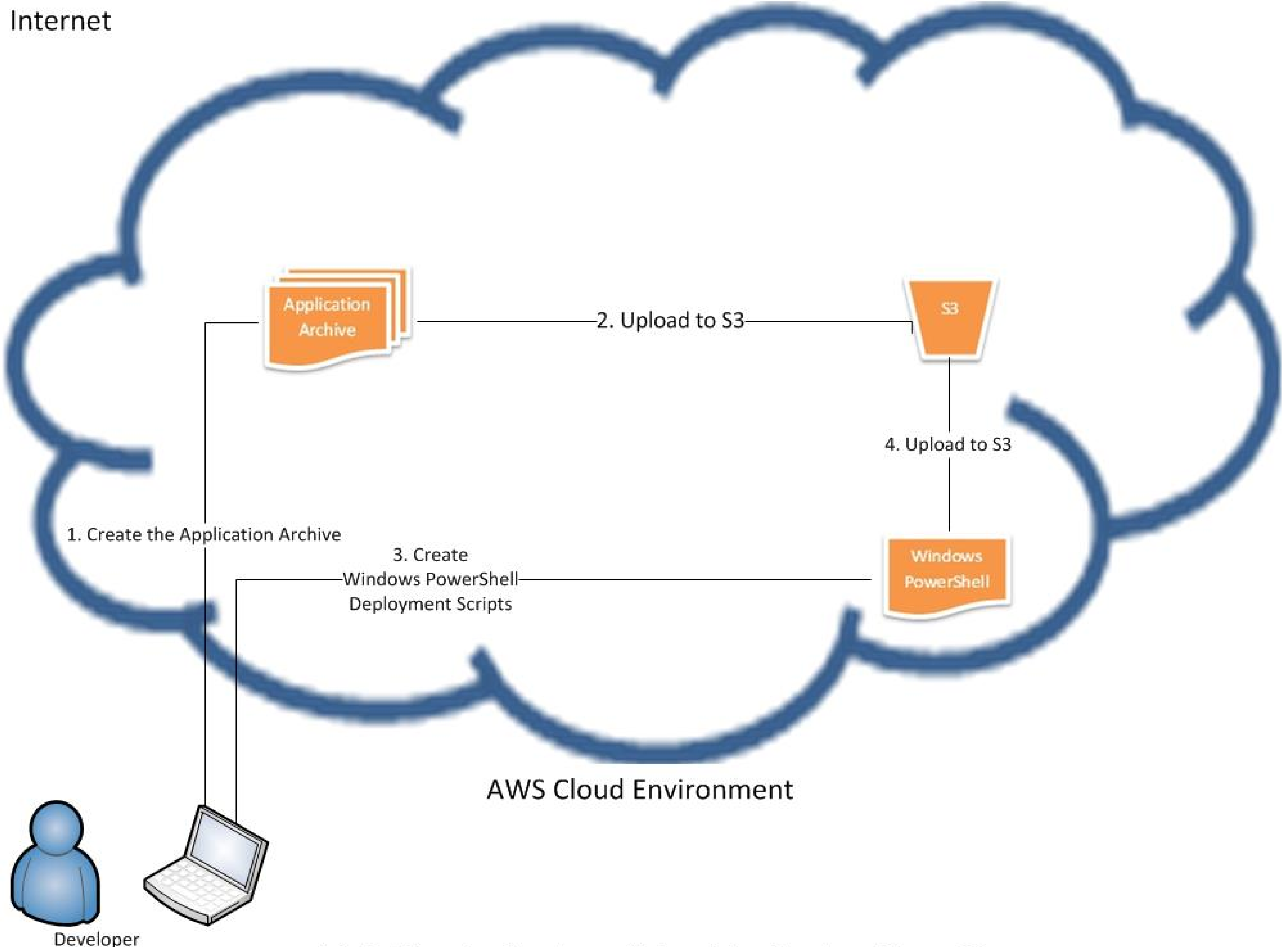


JobBuilder Application - New AWS CloudFormation Templates

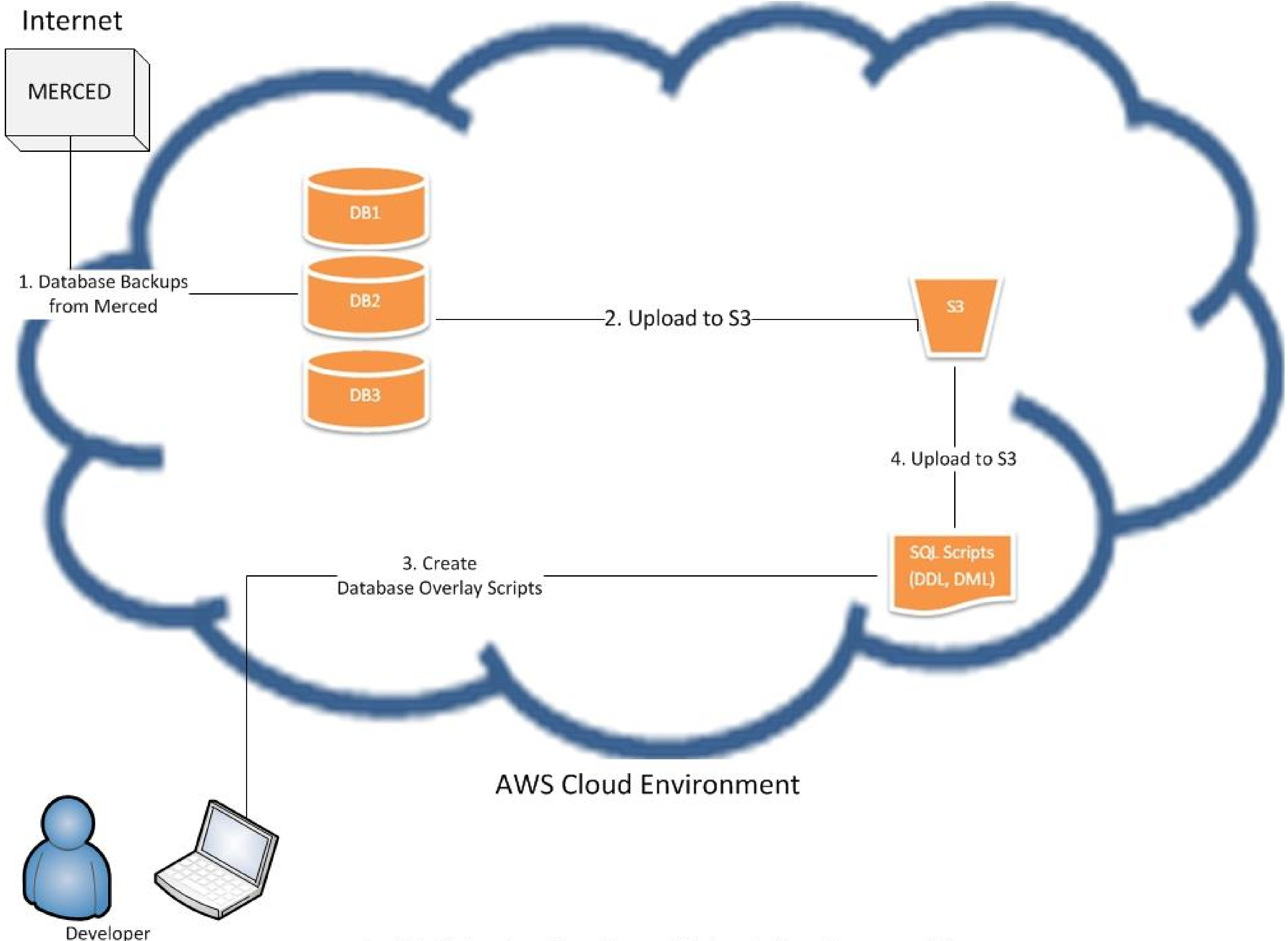


{ } ec2.template.json

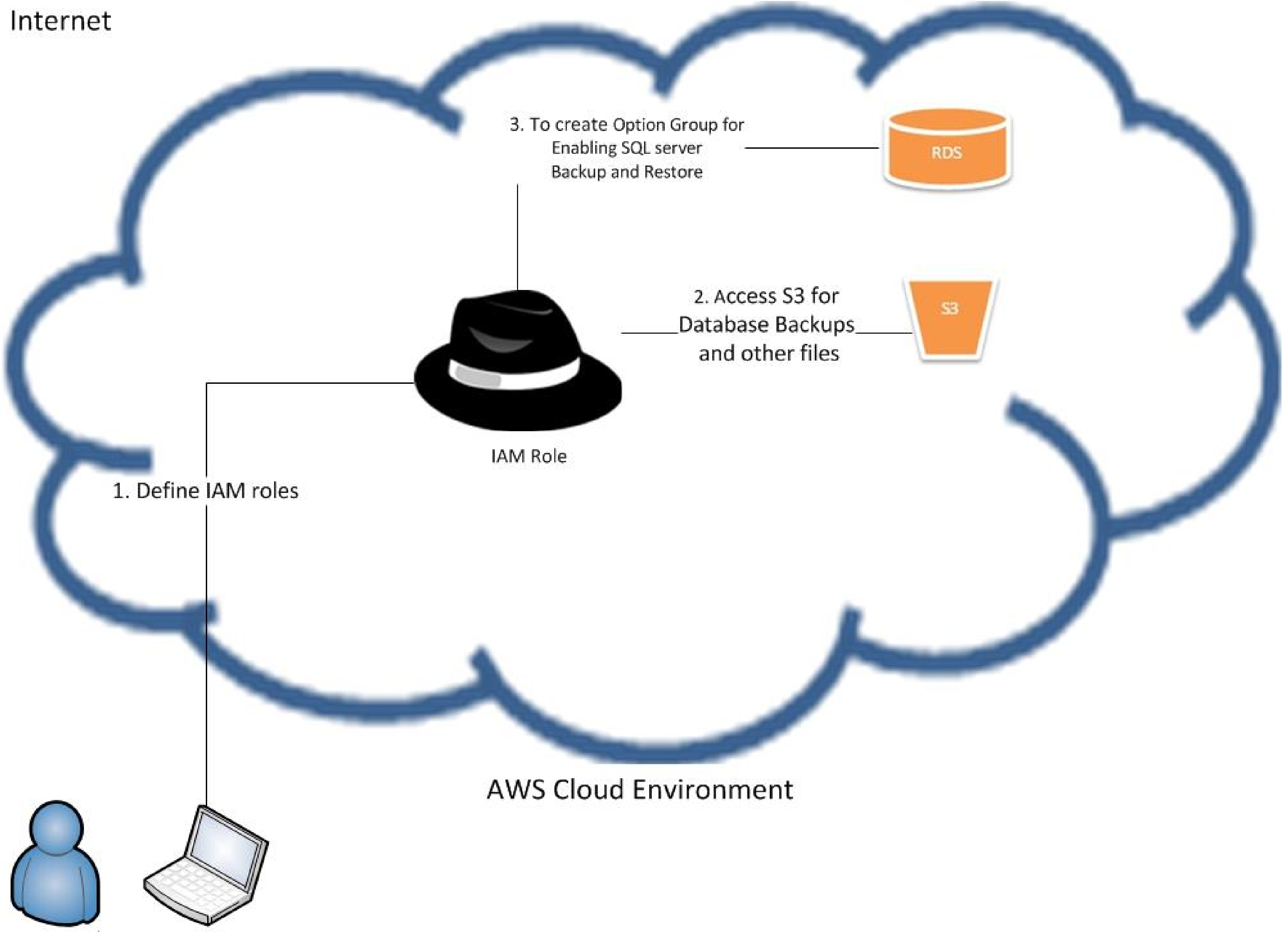
```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "Creates a load balanced and auto scaled web site running under IIS. It uses a SQL Server database. It
  "Parameters" : {
      "AppZipName" : {
      "Description" : "Code version to be deployed. This doubles as the key of the zip file with deployed files in the S
      "Type" : "String",
      "Default" : "1.0"
    },
      "WebsiteDomain" : {
      "Description" : "Domain of the web site, using Route 53.",
      "Type" : "String",
      "Default" : "1ucoptest.com"
    },
    "CertArn" : {
      "Description" : "Certificate ARN",
      "Type" : "String"
    },
    "KeyName" : {
      "Description" : "The EC2 Key Pair to allow RDP access to the instances",
      "Type" : "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair.",
      "Default" : "Ucop-west-key"
    },
    "DbServerAddress" : {
      "Description" : "DB server address",
      "Type" : "String"
    },
    "DBJobBuilderUser": {
      "Description": "JobBuilder database user name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription": "must begin with a letter and contain only alphanumeric characters.",
```



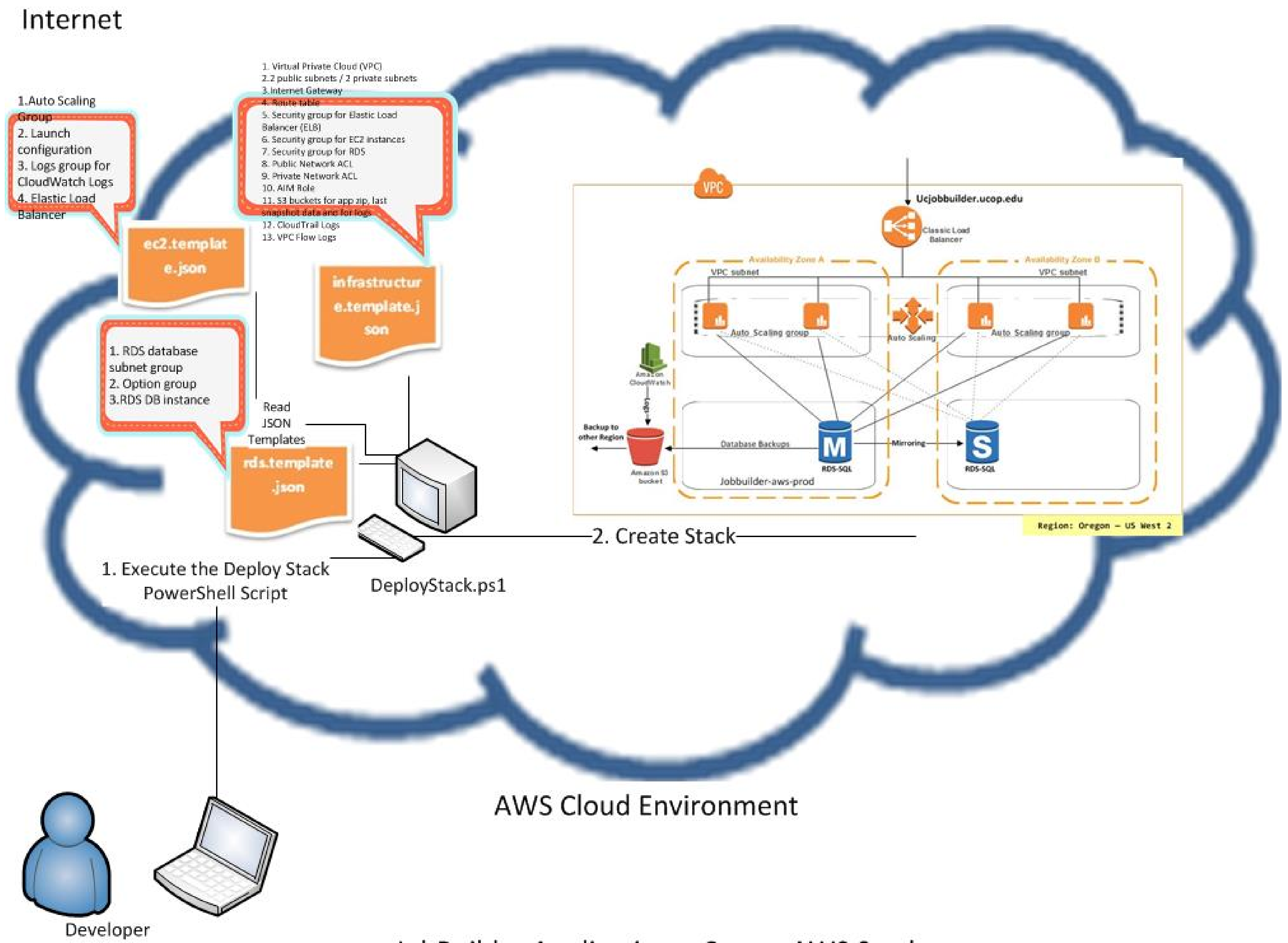
JobBuilder Application – Upload Application Bits to S3



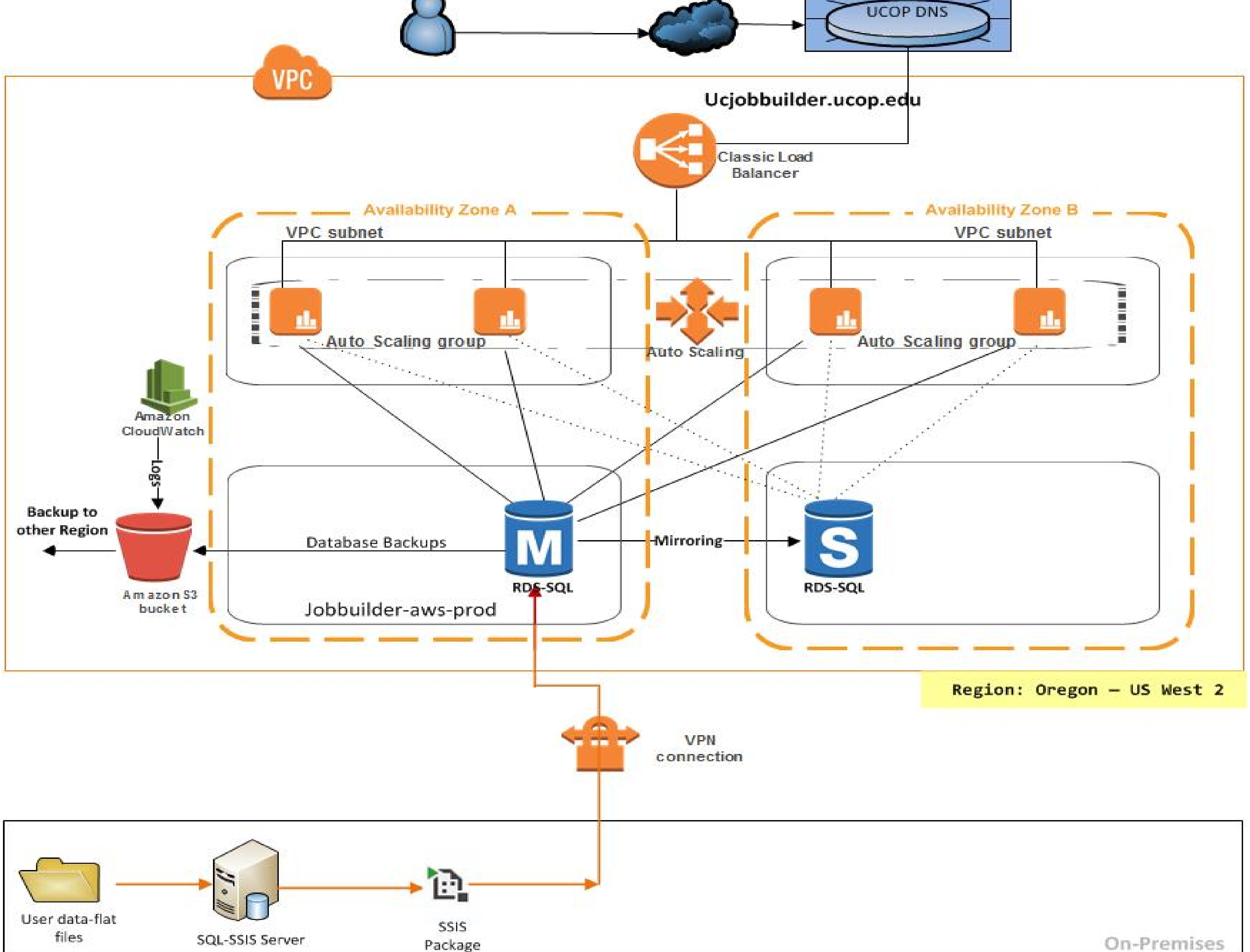
JobBuilder Application – Upload Database to S3


Internet
3. To create Option Group for Enabling SQL server Backup and Restore
RDS
2. Access S3 for Database Backups and other files
S3
IAM Role
1. Define IAM roles
AWS Cloud Environment
Developer

JobBuilder Application – Define IAM Roles



JobBuilder Application – Create AWS Stack


UCOP DNS
VPC
Ucjobbuilder.ucop.edu
Classic Load Balancer
Availability Zone A
Availability Zone B
VPC subnet
VPC subnet
Auto Scaling group
Auto Scaling
Auto Scaling group
Amazon CloudWatch
Logs
Backup to other Region
Amazon S3 bucket
Database Backups
M
RDS-SQL
Mirroring
S
RDS-SQL
Jobbuilder-aws-prod
Region: Oregon – US West 2
VPN connection
User data-flat files
SQL-SSIS Server
SSIS Package
On-Premises

Shibboleth Integration

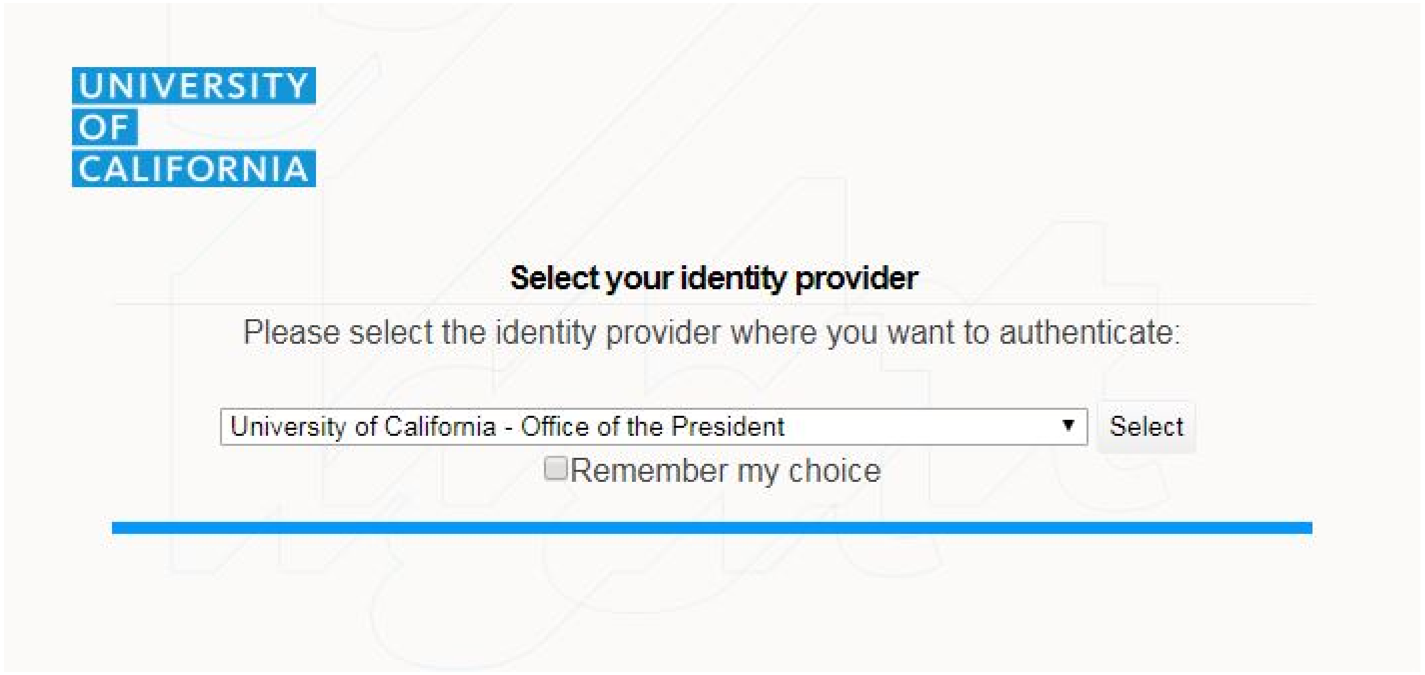
The Job Builder application is integrated with the campus Shibboleth IDP for proper authentication.

The Job Builder application serves as a separate Service Provider.

The following is how the Authentication is handled between UC Job Builder and campus IDP.

- User Navigates to UC JobBuilder
- UC JobBuilder sends user to their campus IdP to authenticate
- IdP redirects back to the UC Job Builder (with attributes)
- UC Job Builder authorizes user and sends user to the home page

Shibboleth Login Page

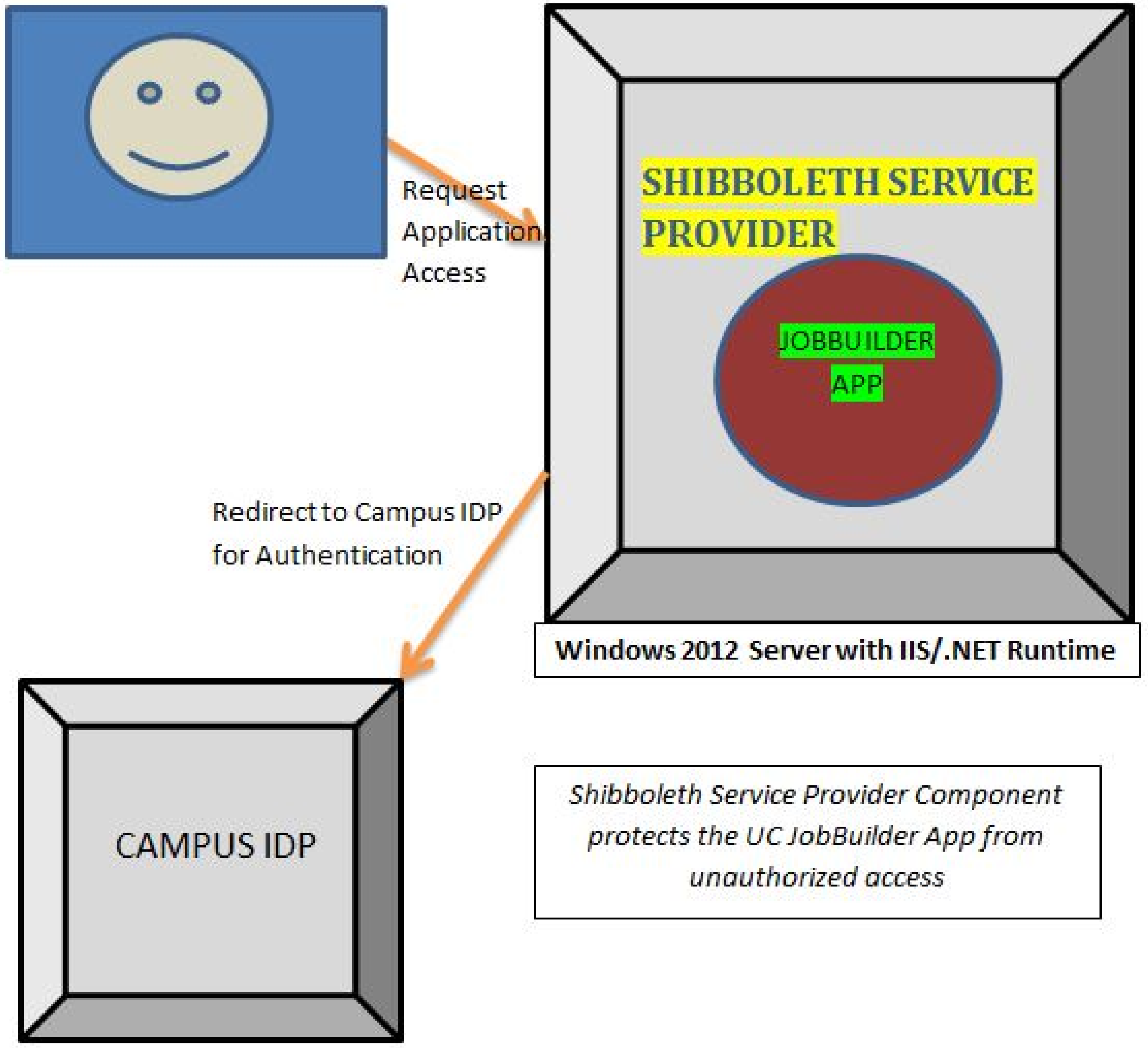


Select your identity provider

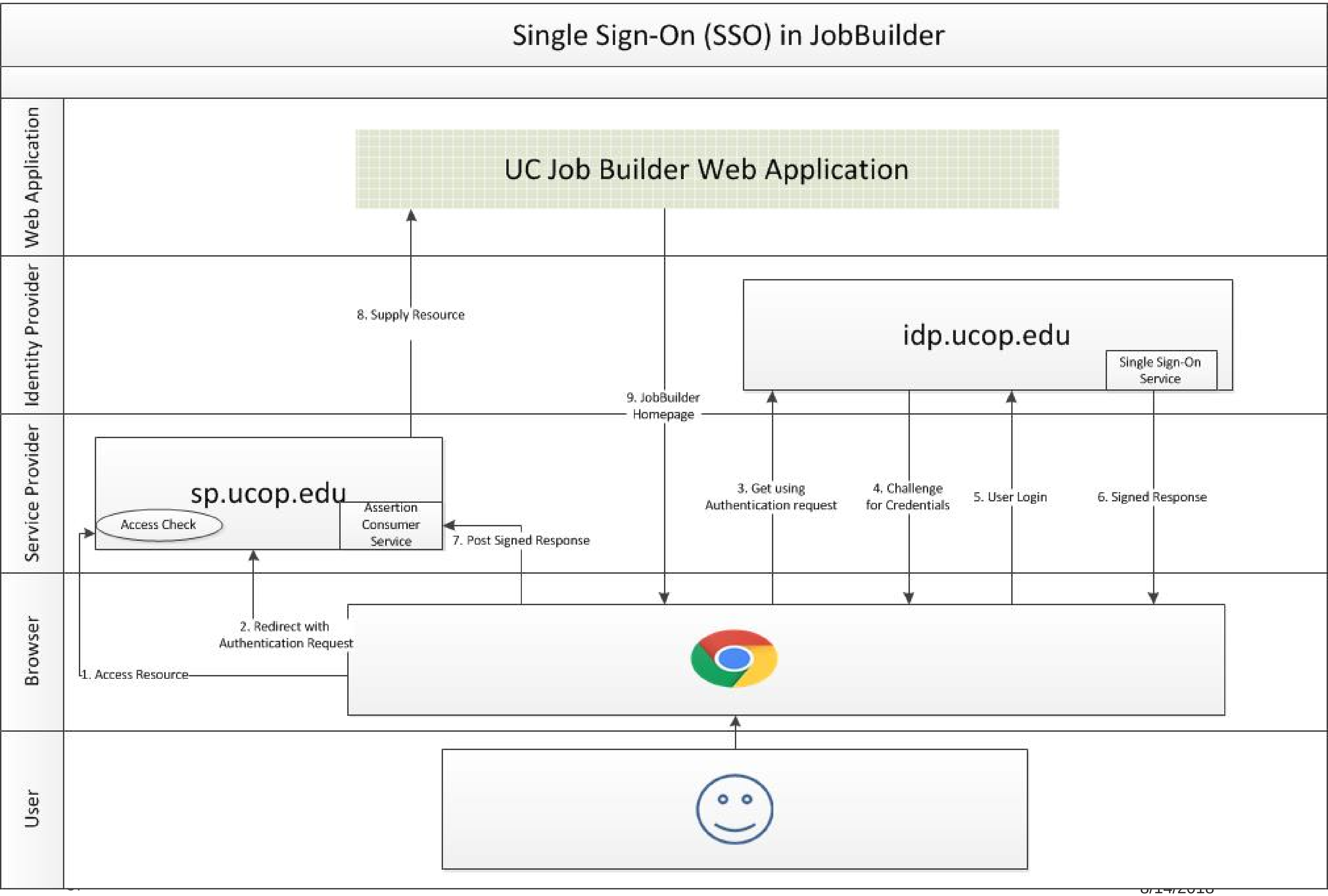
Please select the identity provider where you want to authenticate:

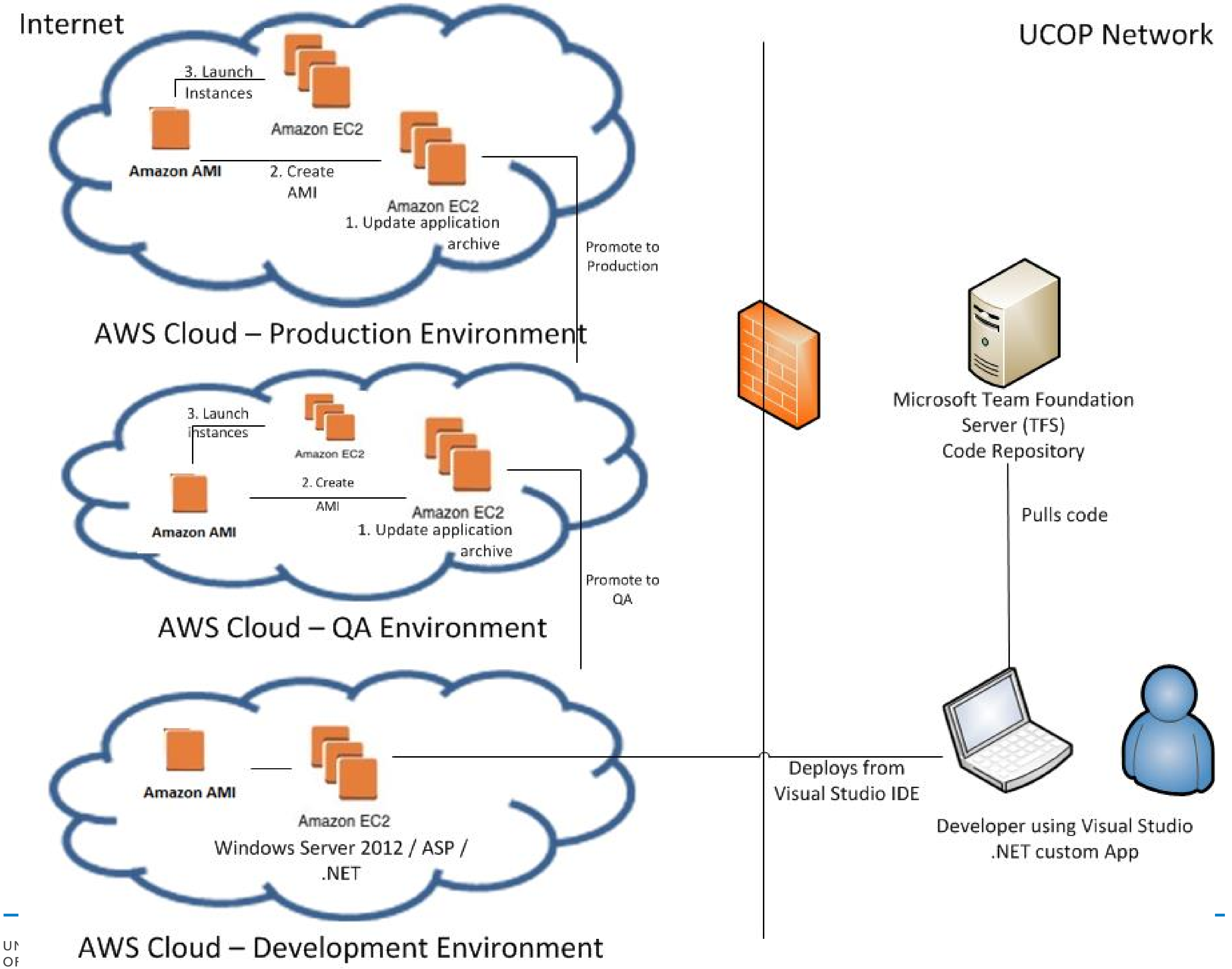
University of California - Office of the President ▾ Select

☐ Remember my choice

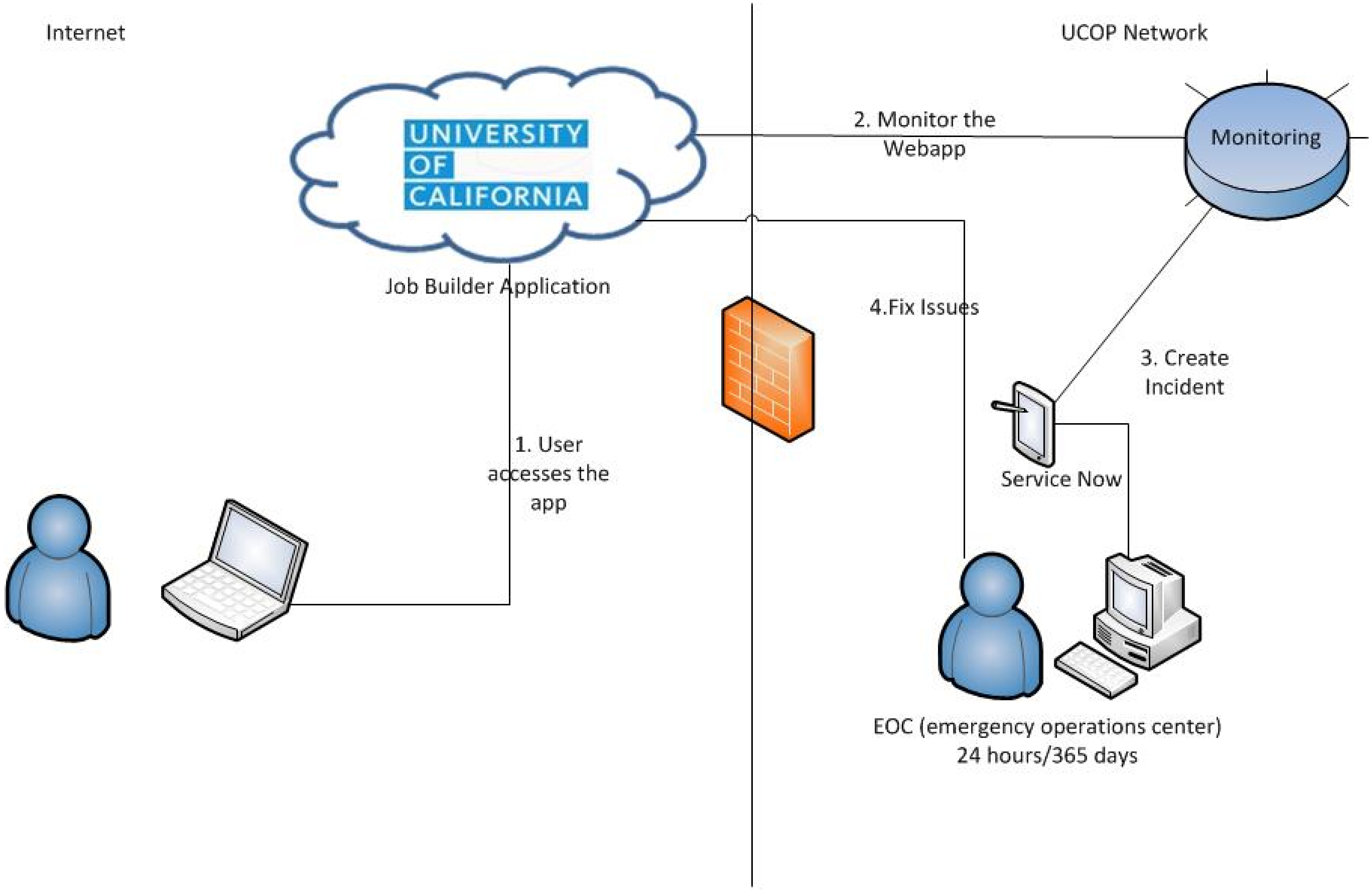

Request
Application
Access
SHIBBOLETH SERVICE PROVIDER
JOBBUILDER APP
Windows 2012 Server with IIS/.NET Runtime
Redirect to Campus IDP for Authentication
CAMPUS IDP

Shibboleth Service Provider Component protects the UC JobBuilder App from unauthorized access

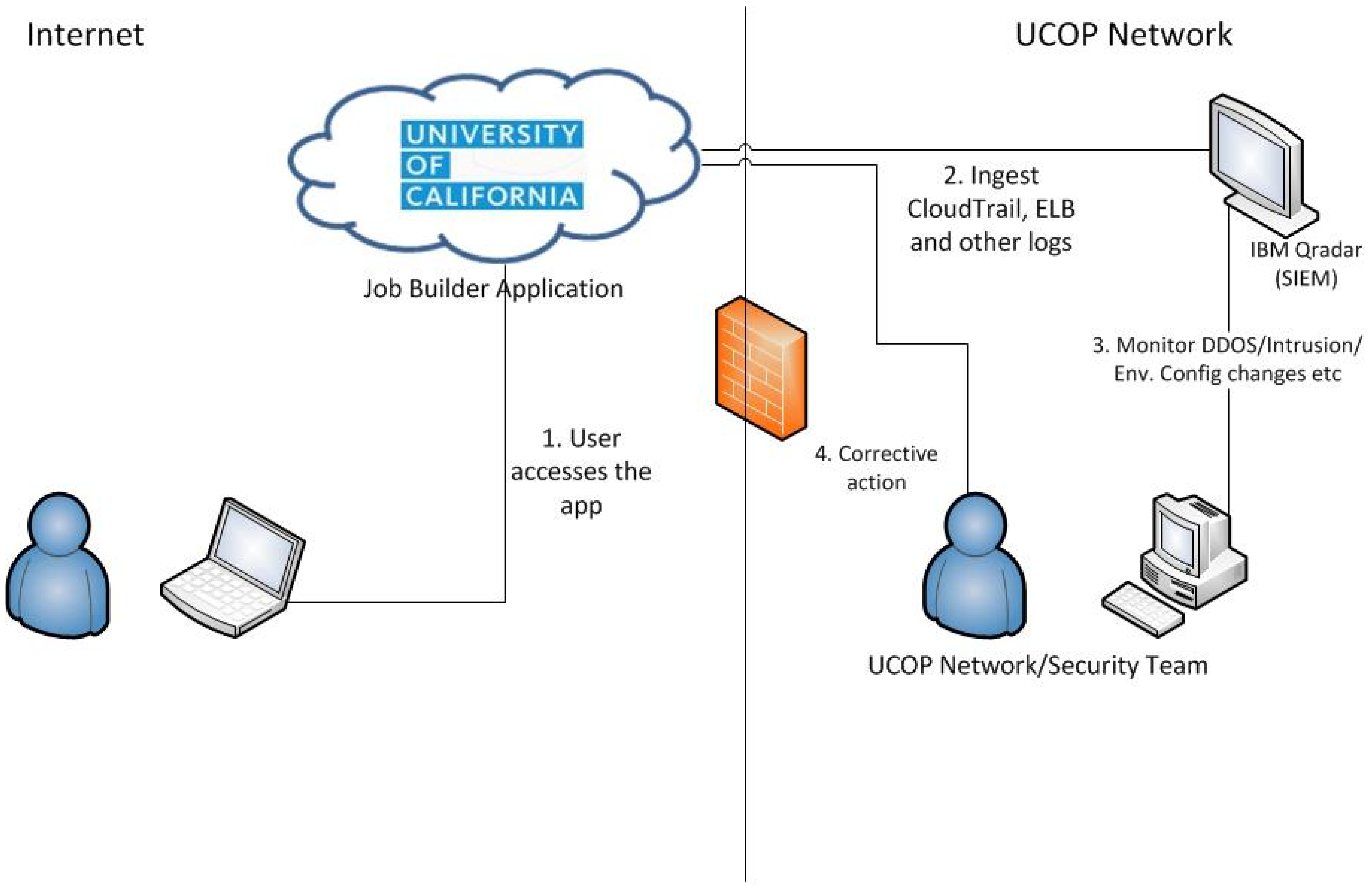

Single Sign-On (SSO) in JobBuilder
Web Application
Identity Provider
Service Provider
Browser
User
UC Job Builder Web Application
idp.ucop.edu
Single Sign-On Service
sp.ucop.edu
Access Check
Assertion Consumer Service
1. Access Resource
2. Redirect with Authentication Request
3. Get using Authentication request
4. Challenge for Credentials
5. User Login
6. Signed Response
7. Post Signed Response
8. Supply Resource
9. JobBuilder Homepage


Internet
UCOP Network
3. Launch Instances
Amazon EC2
Amazon AMI
2. Create AMI
Amazon EC2
1. Update application archive
Promote to Production
AWS Cloud – Production Environment
3. Launch instances
Amazon EC2
2. Create AMI
Amazon AMI
Amazon EC2
1. Update application archive
Promote to QA
AWS Cloud – QA Environment
Microsoft Team Foundation Server (TFS) Code Repository
Pulls code
Amazon AMI
Amazon EC2
Windows Server 2012 / ASP / .NET
Deploys from Visual Studio IDE
Developer using Visual Studio .NET custom App
AWS Cloud – Development Environment

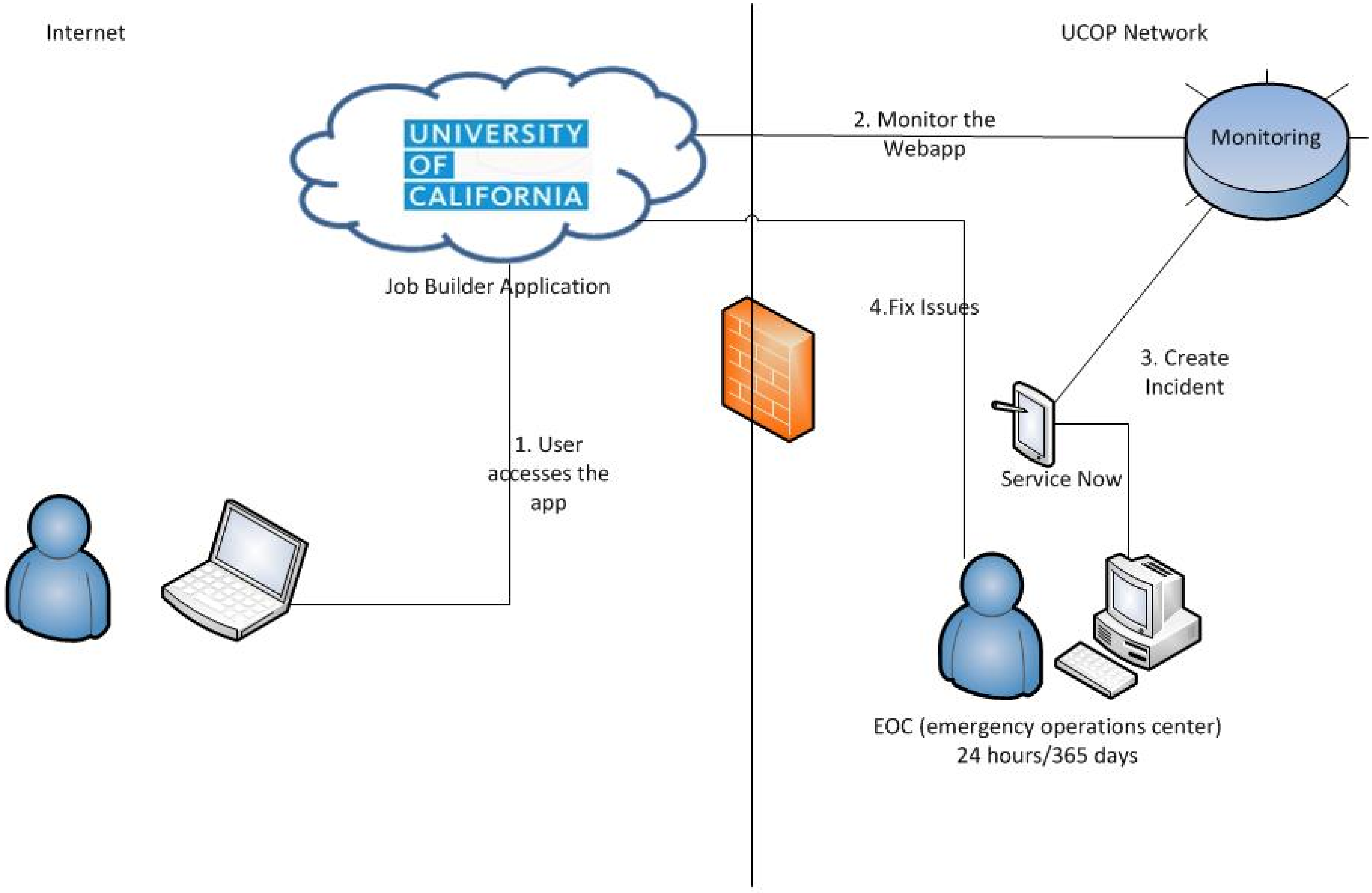
JobBuilder Application Deployment Model – Build creation and deployment



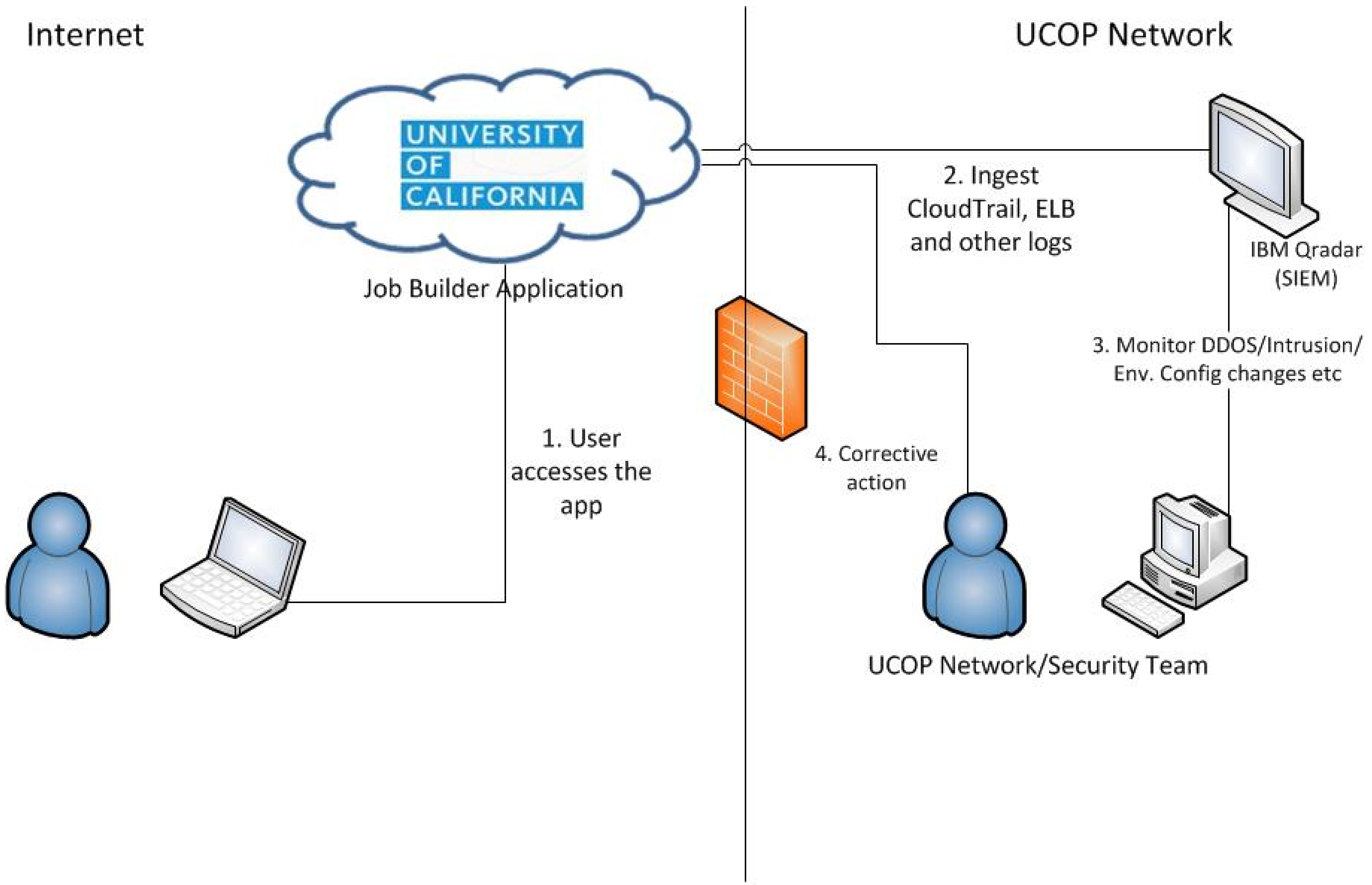
Monitoring: Integrated workflow – Incident creation in ServiceNow, notify EOC



Security Information and Event Management: Collecting and analyzing security logs in real-time



Monitoring: Integrated workflow – Incident creation in ServiceNow, notify EOC



Security Information and Event Management: Collecting and analyzing security logs in real-time

System Requirements

- Fine grain control of the environment – EC2 and the various server/network components
  - [Solution] CloudFormation Script using JSON, EC2 instances

- Environment should be scripted - so setup and tear down is easy
  - [Solution] CloudFormation Script using JSON

- Comply with the network and security needs – PROPER CONTROLS
  - [Solution] VPC, IAM, Security Groups (Virtual Firewall), network access control list (ACL), Cloudtrail, Cloudwatch (monitoring), storing logs on S3.

- Application Alerting and EOC integration
- [solution]
  - Cloudwatch (monitoring) – Alerts us of any failures
  - internal UCOP monitoring – alerts when website is down
    - alerts EOC
    - creates incidents in service-now

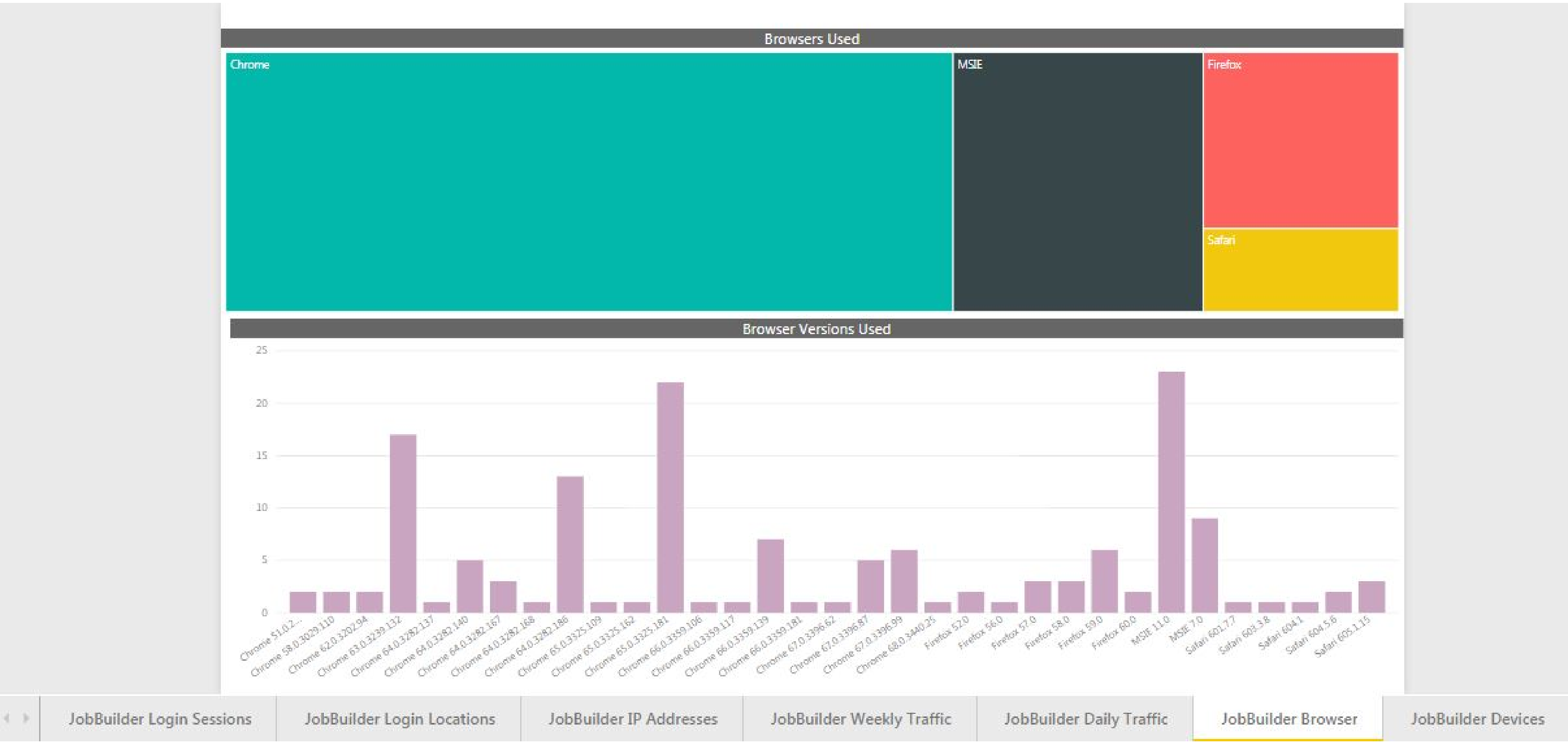
- Secure and Controlled access to the environment
  - [solution] VPC, VPN, only infrastructure group has access to prod

- security information and event management (SIEM) Integration
  - [solution] Integration witH qradar in real time

New System on AWS

- AWS PaaS (Platform-as-a-Service)
- Reliable > 99 % for most AWS services
- Productivity gains (faster time to market)
- Elastic and easily Scalable (add & subtract resources)
- Secure global computing Infrastructure
- Multi-AZ environment for disaster recovery
- Cost-Effective (on demand compute and storage)
- AWS VPC, EC2, IAM, RDS, ELB, S3, Security Groups (Virtual Firewall), Cloudtrail, Cloudwatch, others

Data Analytics using MS PowerBI



Problems faced

- EC2 limit in UCOP account. Launching EC2 instance failed.
- We were trying to set up an environment in QA for load testing but the deployment fails due a hard limit of number of EC2 instances allowed per account. We had to make request to increase the limit? Below is the error.
  - “You have requested more instances (6) than your current instance limit of 5 allows for the specified instance type. Please visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to this limit. Launching EC2 instance failed.”
- Convert ELB logs to JSON format for Qradar
- Cloudwatch logs are available to QRadar
- AWS Toolkit unable to deploy from Visual Studio – unsupported version

Future

- Other UC campuses can adopt Job Builder for their campus
- Use more AWS cloud native services (preferably SaaS)
- Implementation Web Application Firewall to provide critical security controls to protect the web app - strong, adaptable security
- Use reserved instances and bring your own license (BYOL) to further reduce the cost of operations

Team Credits

- Technical Sponsor: Donna Yamasaki
- Business Sponsor: Brad Chilcoat
- Development Team: Charles Shook, Bret Dowell, Rajani Prakash, Dimitry Braverman
- Project Manager: Udette Flesch
- Infrastructure Sponsors: Kevin Cornish, Gilbert Loo, Timothy Hanson, Erik Freitag, Mark Cruz
- Infrastructure Team: Jason Lam, Ashley Gould, Mark Boyce, Jim Kassenbrock, Daniel Adeniji, Sandy Owyang, Samer Khalil, Justin Tipton, Mario Nakane
- Data Visualization Intern: Devin Pon (UC Davis)
- AWS Solutions Architect: Sushant Prasad

Questions ?

Thank You

Contacts:

- Bret.Dowell@ucop.edu
- Rajani.Prakash@ucop.edu
- Sushant.Prasad@ucop.edu