
Cloud and Big Data

Assignment Project Exam Help

Sambit Sahu

IBM Research

<https://powcoder.com>

Add WeChat powcoder



Course Objective

- **Graduate level course on Cloud Computing**

- Focus is on learning and building extremely large scale systems and applications leveraging Cloud
 - Building blocks and design patterns in designing backend of typical Internet Scale application
- Learn concepts as well as hands-on experience by using real cloud and cloud technologies.
- Three key objectives: learn how to use a cloud, leverage cloud to build applications, build scalable intelligent systems

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- **We shall learn cloud technologies by using real clouds and services - Amazon AWS, Google Cloud, Hadoop/Spark, Kafka, Elastic, Dynamo etc.**

- **Required background**

- Programming experience with one of the following Java/Python, web services basics
- Operating Systems concepts, networking concepts would help you understand more

--> If you are not familiar with web services, take a look at materials on any web application design technologies.

What would you learn in this course...

- HowTo

- How to use a Cloud as a compute node?
- How to use cloud to design an Internet scale application?
- How to process a very large amount of data?
- How to build your own cloud using open source?

Assignment Project Exam Help

- Concepts: Building Blocks

- Virtualization, Containers, Serverless
- Peta-byte scale storage systems
- Event and messaging systems (Kafka)
- noSQL datastore (Cassandra, mongo, DynamoDB,...)
- Elastic Search
- Compute in a cluster
- Intelligent AI applications
- ...

<https://powcoder.com>

Add WeChat powcoder

- Case studies with real systems/cloud

- Compute Cloud, Storage Cloud, Data Cloud

■ Cloud Platform and Programming

- Basic cloud concepts
- Hands-on experience with Amazon AWS Cloud
- Virtualization as an enabling technology
- Virtualization vs Containers vs Serverless
- ***Build a Web application leveraging cloud***

■ Building Blocks in an Extremely Large Scale Application

- Scalable data store and noSQL database
- Message Queues: Kafka
- Unstructured data and queries: Elastic Search
- In-memory data store
- devOps: Containers, micro-services, logging and monitoring
- ***Build a scalable application using scalable, event-driven pattern***

■ Private Cloud

- Understand key concepts for building a cloud
- Use Openstack cloud management stack
- devops/chef/puppet for private cloud automation
- ***Build your own cloud***

■ Big Data Computing Platform and Programming

- Hadoop eco-system, and batch data processing & storage
- MapReduce, Hive, Hbase
- *Spark and Spark Streams*
- ***Intelligent Real-time system design using Spark***

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Tentative Syllabus/Lectures

- Intro to Cloud: IaaS, PaaS, SaaS cloud, AWS, GCP, Azure Cloud (but we focus on building using AWS)
- Designing a web application using cloud
- Virtualization as Cloud Enabling technology; Virtualization vs Containers
- Building Private Cloud (OpenStack)
- DevOps in a Cloud and Micro-services Architecture
- Designing Extremely Large Scale Applications
 - Message Queue (Kafka)
 - Event Notification
 - Scalable no-SQL
 - Lambda architecture
 - Indexing and searching unstructured data (Elastic Search)
- Computing in a Cluster
 - Hadoop/MR
 - Spark based compute model
- Use cases: Designing Intelligent Services in a Cloud - we will use a variety of AWS ML and Google ML APIs to design interesting use case

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Tentative Course Schedule

Date	Topic	Reading List
09/08	Intro to Cloud	
09/15	Cloud Programming	GFS
09/22	Designing Scalable Web Application	BigTable
09/29 [A1]	Designing Web Scale Applications	Kafka
10/06	Message Queues and Logs	Cassandra, DynamoDB
10/13	noSQL database, Elastic Search	MapReduce
10/20 Quiz1 [A2]	Containers, Kubernetes, Devops	anthos
10/27	Cluster Computer: Spark	spark
11/03	Spark Data Frames	Borg
11/10 [A3]	Spark Advanced	spanner
11/17	Private Cloud	
11/24	Intelligent Systems	
12/01 Quiz 2	Advanced Topics	
12/08	Advanced Topics	
12/15	Final Demo	

Course Material

- **Lecture Notes**

- Each lecture will have a theme topic. Lecture slides will be provided for each lecture. Additional reference materials will be specified.

- **Reading List**

- A set of landmark papers in the area of large scale systems
- You submit a paper summary by answering the provided questions.

Assignment Project Exam Help

<https://powcoder.com>

- **Three programming Assignments**

- **A final Course project**

Add WeChat powcoder

- **Reference Texts**

- AWS in Action
- Elastic Search in Action
- Kafka Definitive Guide
- Hadoop: The Definitive Guide
- Learning Spark

Grading and requirements

- 2 Quizzes -- 25%
- Assignments – 35% grade
 - 3 homework stressed on technologies and programming
- Course project -- 40% grade
 - Students may team up
- Submission process – everything to be done using Courseworks and Github

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Project: Learn how to innovate in this space

- **Objective is to learn how to innovate in this space**
- **Four phases to your project**
 1. Concept and business idea
 2. Technology viability and architecture
 3. Execution planning and prototyping
 4. Demo, socialization and review
- **Few suggestion**
 - Don't procrastinate – start early. Motivation: Would help you get A+ (and earn millions!)
 - Form your team carefully – asking, interviewing your team mates. Float around some ideas,, kick the tire. Take a look at lot of recent startups that are bought by Google, Apple, FB, Amazon etc. **Take a look at beta.list**
 - Cloud + Social + Mobile is a good recipe for a perfect storm

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

What you need to do soon

- Get account on few popular clouds
 - Amazon AWS (EC2, S3)
 - Google Cloud Platform, Google Storage
 - We are working with Amazon to get free accounts

Assignment Project Exam Help

- Course Project
 - Substantial portion of your grade depends on final course project
 - I will provide a set of project categories that you could choose from or come up with your own. But each project category will have a set of criteria that need to be demonstrated
 - You need to have a team and a project proposal by 02/11/20 5:00pm

<https://powcoder.com>

Add WeChat powcoder

What is Cloud?

- Allows users to request computing/storage resources through web interfaces
- You do not need to own or install or manage these resources.
- Pay as you go - Resources on-demand
- Elastic: Use as much as you want or as less as you want
 - Users can assume infinite amount of compute and storage resources are available.
 - Users can request resources when and what they need and release/remove resources when they don't need.
- Compute and storage resources are now treated as software entities. You get access to such resources programmatically – not by physical hardware anymore!
- So what are the Clouds! Where are the Cloud?
- Read this paper: <http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Why Cloud?

- You can get as many as 1000 machines for an hour for a few dollars to run a complex application!
- You don't need to manage, maintain or fix any machines!
- You can use as little as 1 machine or as many as 10000 machines depending on what your current needs are!
- Two key focus: on-demand and elastic!

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Essential Characteristics

- *On-demand self-service.* A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- *Broad network access.* Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- *Resource pooling.* The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- *Rapid elasticity.* Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- *Measured Service.* Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Service Models

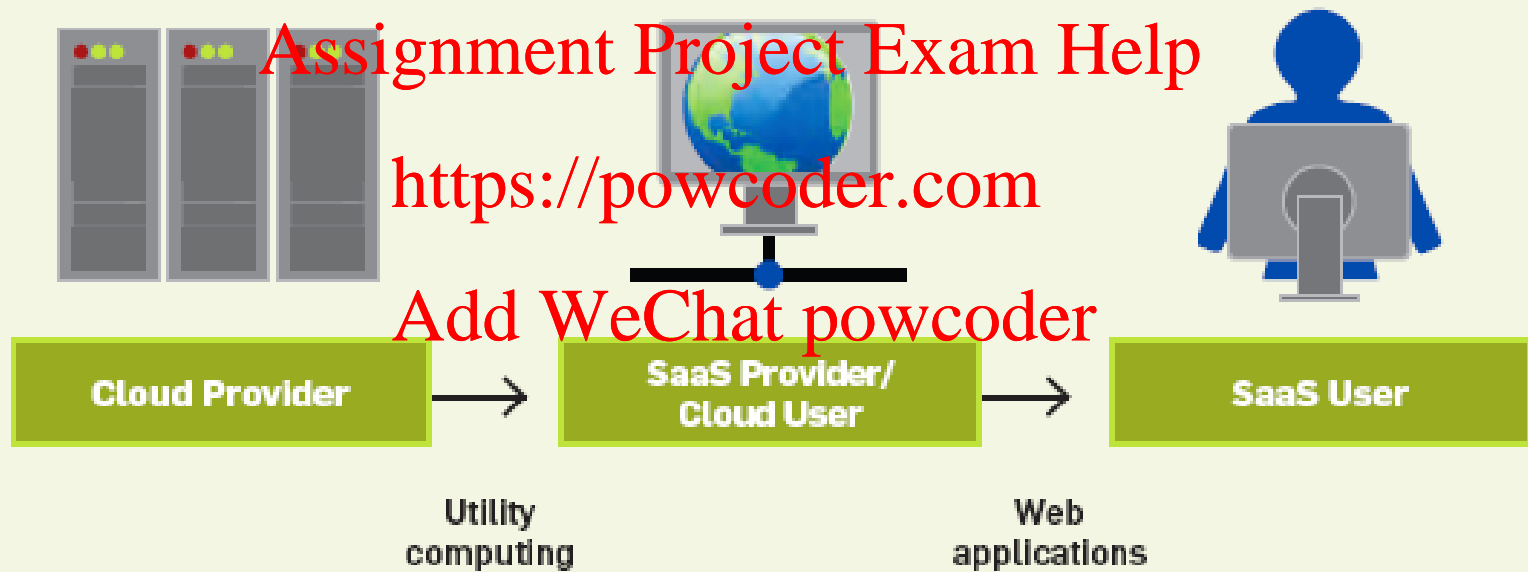
- *Cloud Software as a Service (SaaS)*. The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- *Cloud Platform as a Service (PaaS)*. The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.
- *Cloud Infrastructure as a Service (IaaS)*. The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models

- *Private cloud.* The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.
- *Community cloud.* The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.
- *Public cloud.* The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- *Hybrid cloud.* The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

Berkeley View of Cloud Definition

Figure 1. Users and providers of cloud computing. We focus on cloud computing's effects on cloud providers and SaaS providers/cloud users. The top level can be recursive, in that SaaS providers can also be a SaaS users via mashups.



- IaaS → SaaS Provider → SaaS User

Source: Above the Clouds: A Berkeley View of Cloud Computing

Different types of utility model

- IaaS Cloud (Amazon EC2)
 - Low level of computing resource abstraction
 - Provides a (virtual) machine to users
 - Makes it hard for IaaS providers to support automatic scaling, failover etc.
- Google AppEngine
 - Targeted at web applications
 - Enforces an application structure
 - Clean separation between stateless and stateful storage tier
 - Benefit: makes it possible to handle auto-scaling, failover/high availability
- Microsoft Azure
 - Applications need to be written using .NET libraries
 - More flexible than Google AppEngine
 - Able to provide some automated scaling
 - Between Application framework and hardware virtual machines

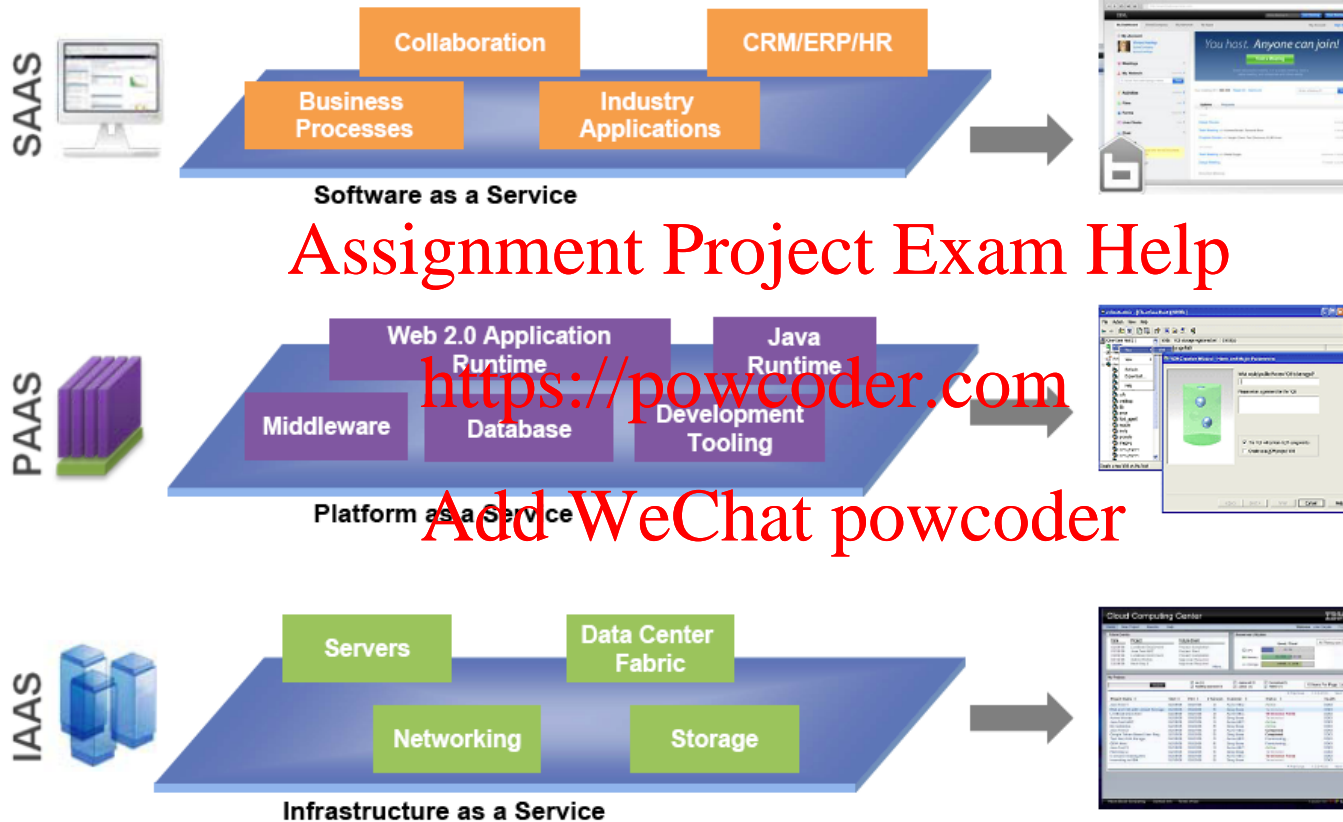
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Different Cloud Offerings: A Layered Perspective

The Layers of IT-as-a-Service



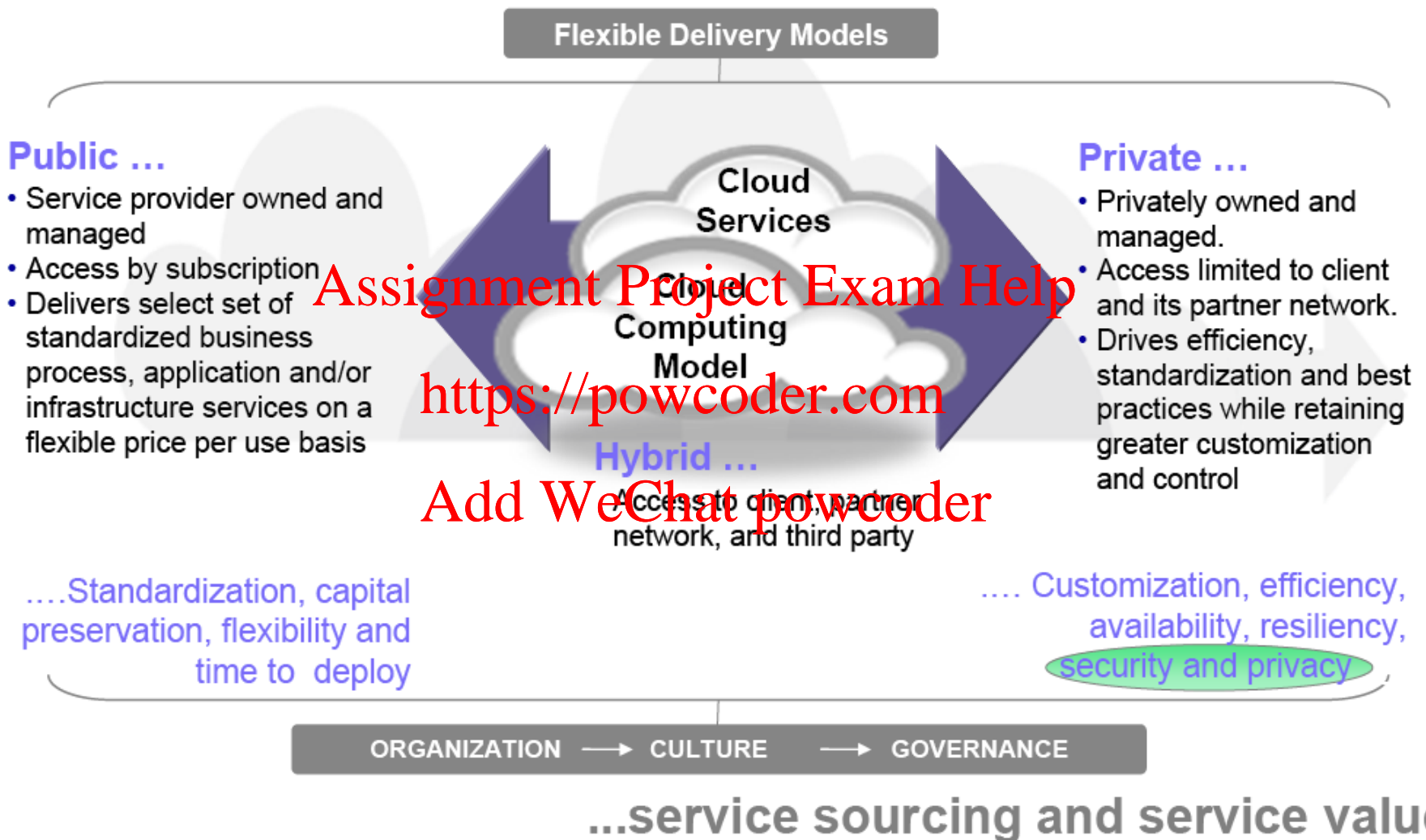
Assignment Project Exam Help

<https://powcoder.com>

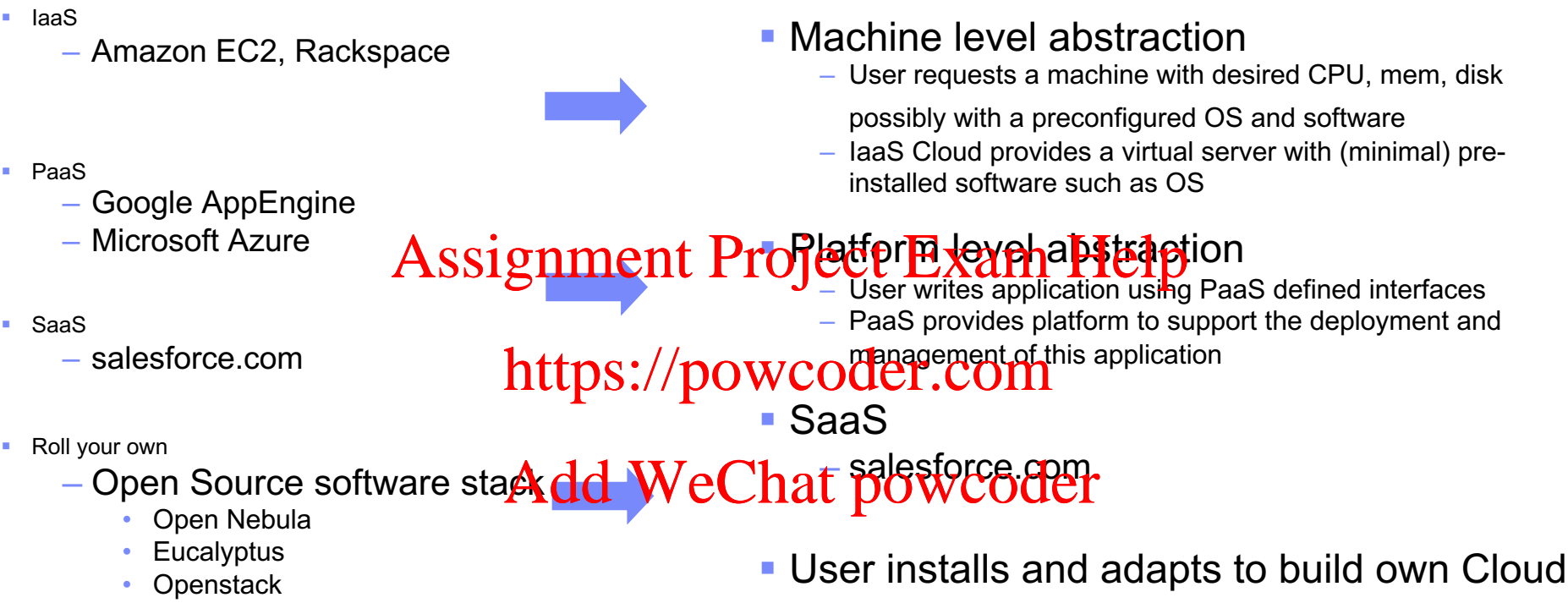
Add WeChat powcoder

- Higher the stack, less control but more automation for user
- Lower the stack, more control but more responsibility for user

Cloud Computing Delivery Models



Example Clouds and Usage Scenario



Cloud Computing Economics

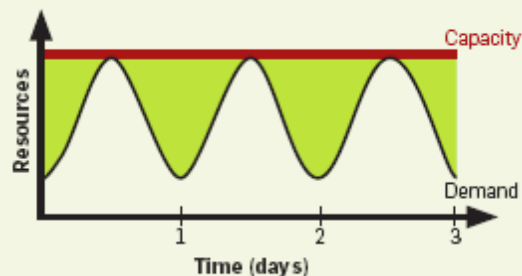
- Three useful usage scenarios
 - Load varying with time
 - Demand unknown in advance
 - Batch analytics that can benefit from huge number of resources for a short time duration
- Why pay-as-you-go model makes sense economically even if costs higher than buying a server and depreciating the h/w
 - Extreme elasticity
 - Transference of risk (of over provisioning)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

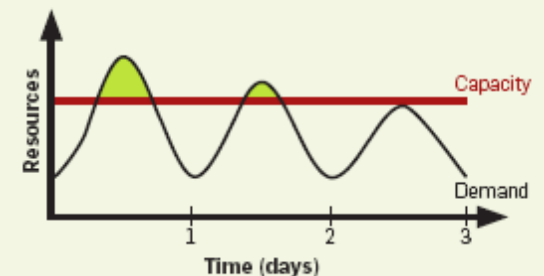
Figure 2. (a) Even if peak load can be correctly anticipated, without elasticity we waste resources (shaded area) during nonpeak times. (b) Underprovisioning case 1: potential revenue from users not served (shaded area) is sacrificed. (c) Underprovisioning case 2: some users desert the site permanently after experiencing poor service; this attrition and possible negative press result in a permanent loss of a portion of the revenue stream.



(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2

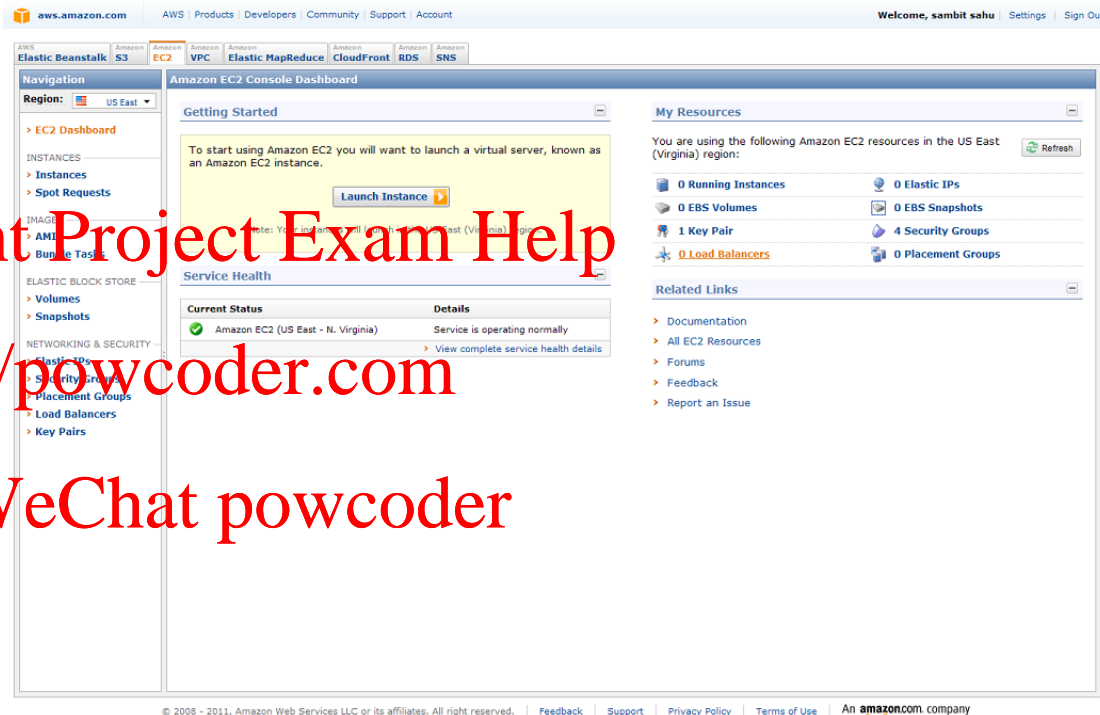
Top obstacles and opportunities for Cloud

Table 2. Top 10 obstacles to and opportunities for growth of cloud computing.

Obstacle	Opportunity
1 Availability/Business Continuity	Use Multiple Cloud Providers
2 Data Lock-In	Standardize APIs; Compatible SW to enable Surge or Hybrid Cloud Computing
3 Data Confidentiality and Auditability	Deploy Encryption, VLANs, Firewalls
4 Data Transfer Bottlenecks	FedExing Disks; Higher BW Switches
5 Performance Unpredictability	Improved VM Support; Flash Memory; Gang-schedule VMs
6 Scalable Storage	Invent Scalable Store
7 Bugs in Large Distributed Systems	Invent Debugger that relies on Distributed VMs
8 Scaling Quickly	Invent Auto-Scaler that relies on ML; Snapshots for Conservation
9 Reputation Fate Sharing	Offer reputation-guarding services like those for email
10 Software Licensing	Pay-for-use licenses

IaaS Cloud Example: Amazon EC2

- Amazon EC2 provides public IaaS Cloud
- User uses a portal to request a machine with specific resource
 - CPU, memory, disk space
 - Pre-built OS and possibly middleware



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

PaaS Cloud: Google App Engine

- PaaS model
- Provides a platform to host web applications
- App Engine SDK for programming (Python and Java support)
- A set of primitives (datastore, URL fetch, memcache, JavaMail, Images, authentication..)
- User focuses on developing the application in this framework
- Once deployed, scaling, availability etc. are handled by Google App Engine platform

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Let' s use a IaaS Cloud (Amazon EC2)

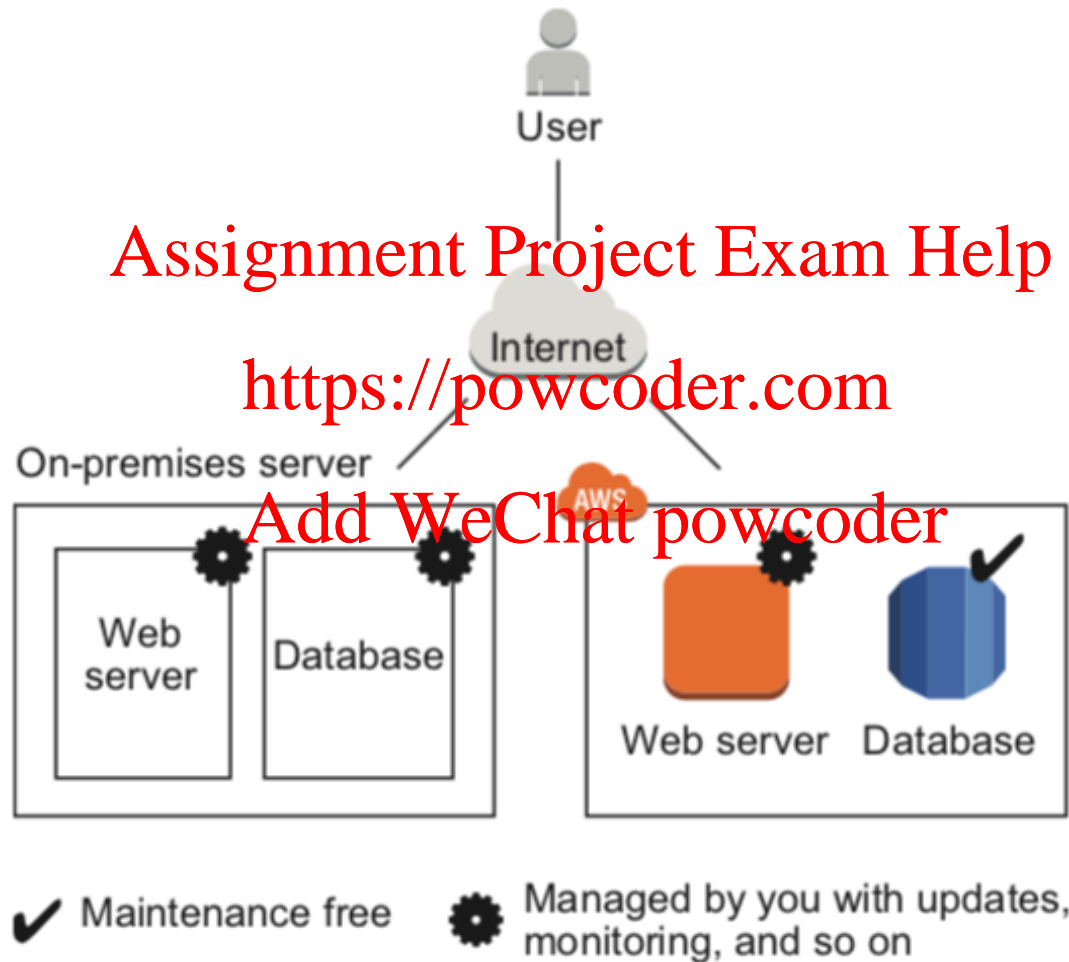
- <http://aws.amazon.com/console/>
- Amazon EC2 console based provisioning demo

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Traditional vs Cloud-based Application



Leveraging Cloud Services to Quickly Build Complex Applications

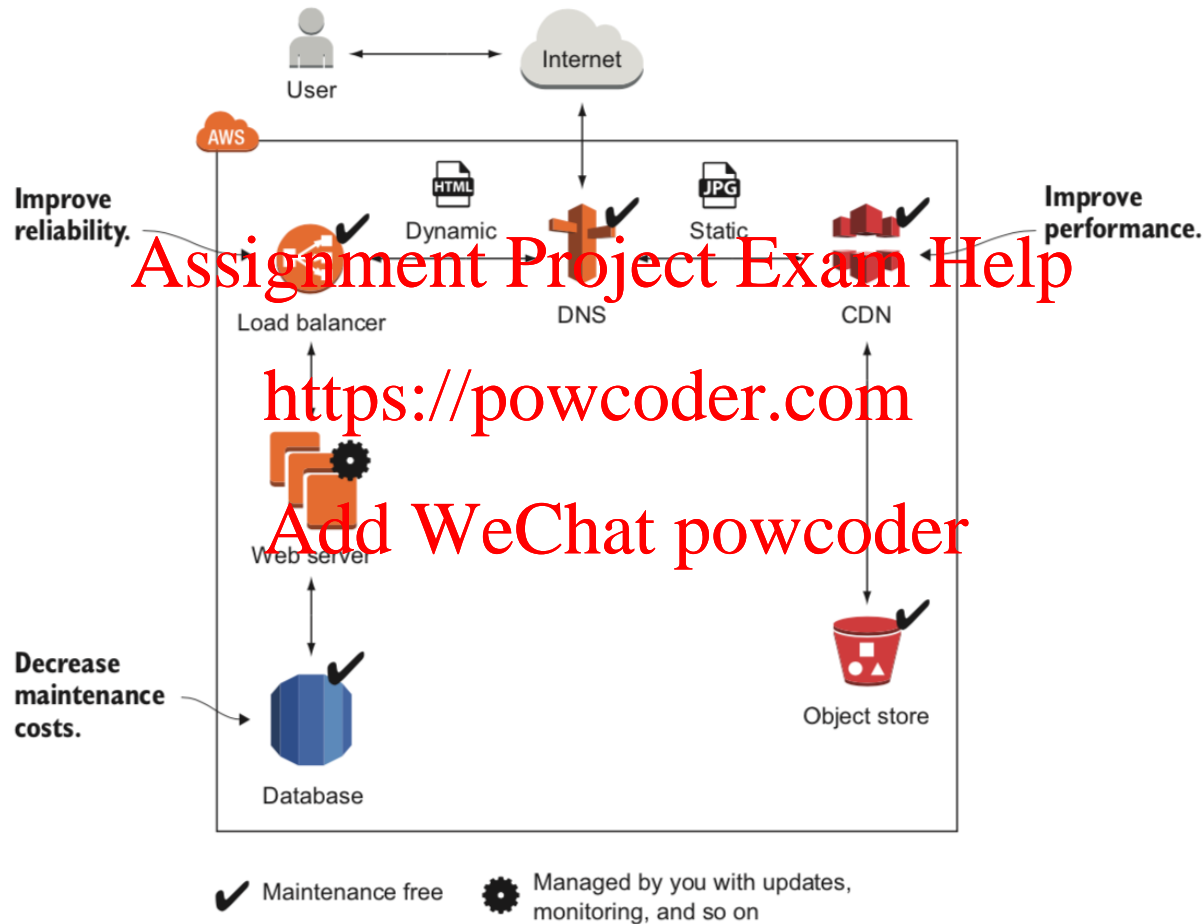
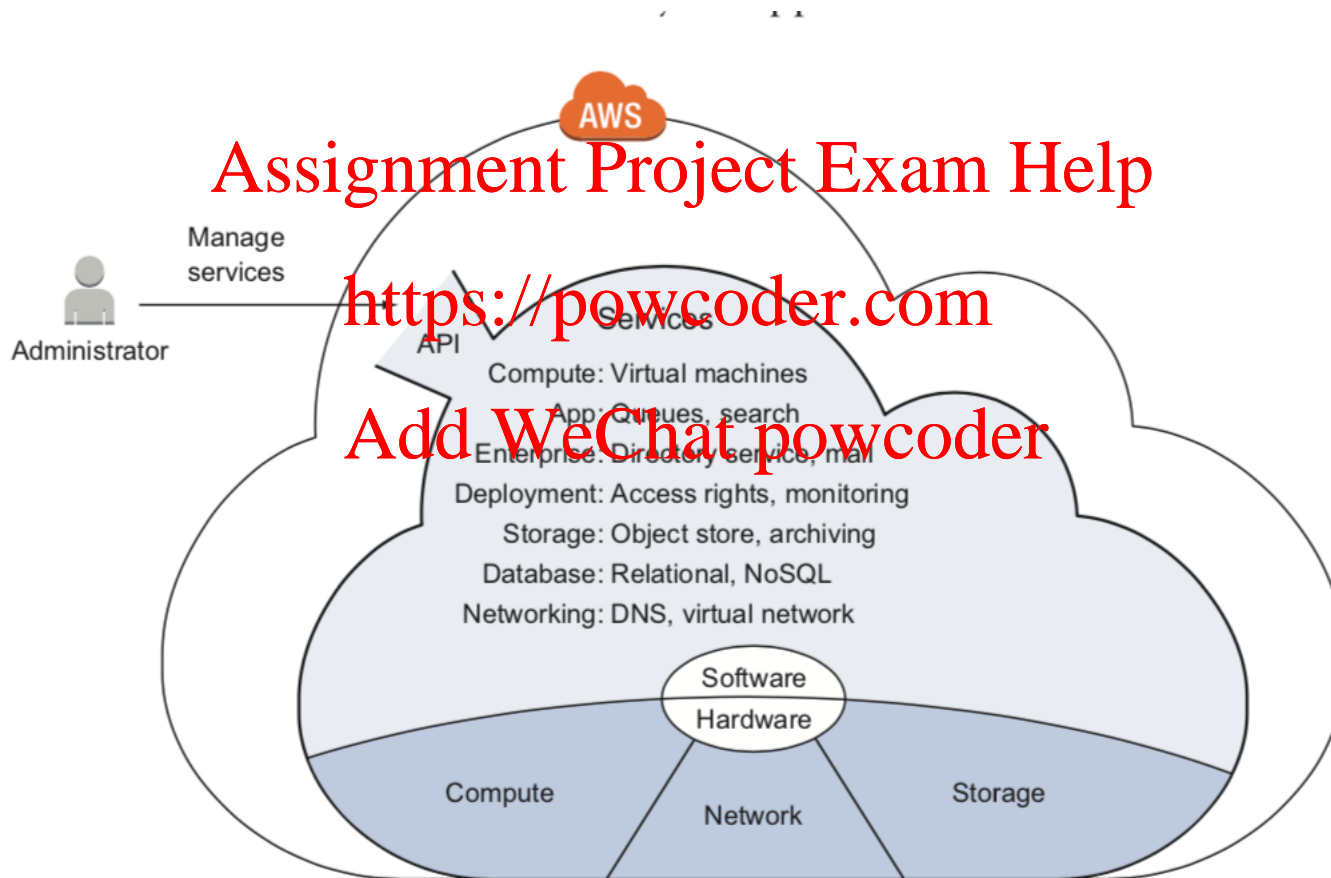
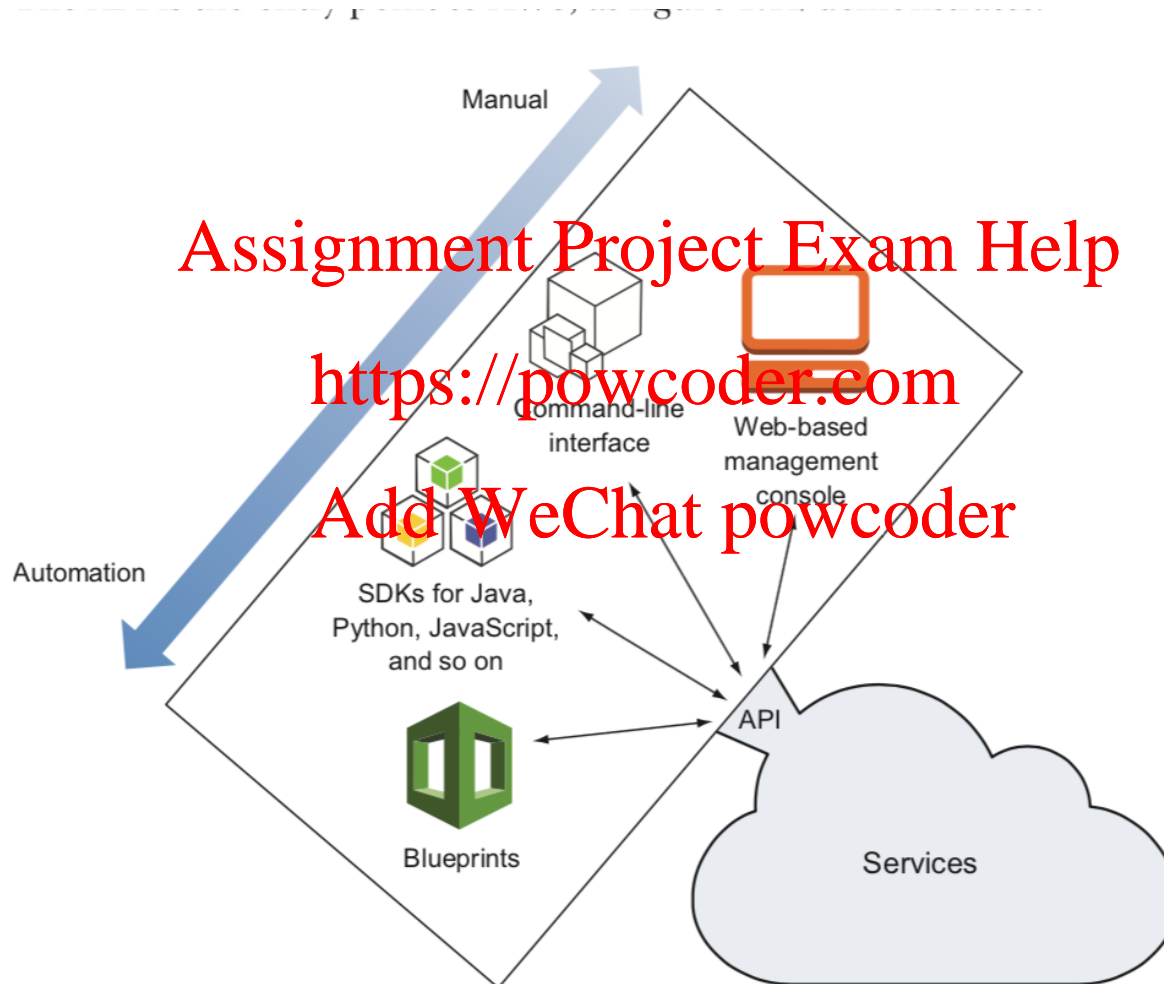


Figure 1.3 Running a web shop on AWS with CDN for better performance, a load balancer for high availability, and a managed database to decrease maintenance costs

Amazon Cloud Services: Accessing through Web APIs



Various Methods to Access AWS



Amazon AWS console (EC2 view)

The screenshot shows the Amazon AWS console interface. At the top, there's a navigation bar with the AWS logo and links for Products, Developers, Community, Support, and Account. The user is logged in as 'sambit sahu'. Below the navigation bar, there's a row of service tabs: Elastic Beanstalk, S3, EC2 (selected), VPC, CloudWatch, Elastic MapReduce, CloudFront, RDS, and SNS. The left sidebar contains a 'Navigation' menu with sections for INSTANCES (Instances, Spot Requests), IMAGES (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots), and NETWORKING & SECURITY (Elastic IPs, Security Groups, Placement Groups, Load Balancers, Key Pairs). The main content area is the 'Amazon EC2 Console Dashboard'. It features a 'Getting Started' section with a 'Launch Instance' button, a 'My Resources' section showing 1 Running Instance, 0 Elastic IPs, 1 EBS Volume, 2 Key Pairs, 0 Load Balancers, 0 Elastic IPs, 0 EBS Snapshots, 6 Security Groups, and 0 Placement Groups, and a 'Service Health' section showing the current status of Amazon EC2 (US East - N. Virginia) as 'Service is operating normally'. A large red watermark is overlaid on the image, reading 'Assignment Project Exam Help' and 'https://powcoder.com'. Below the watermark, it says 'Add WeChat powcoder'.

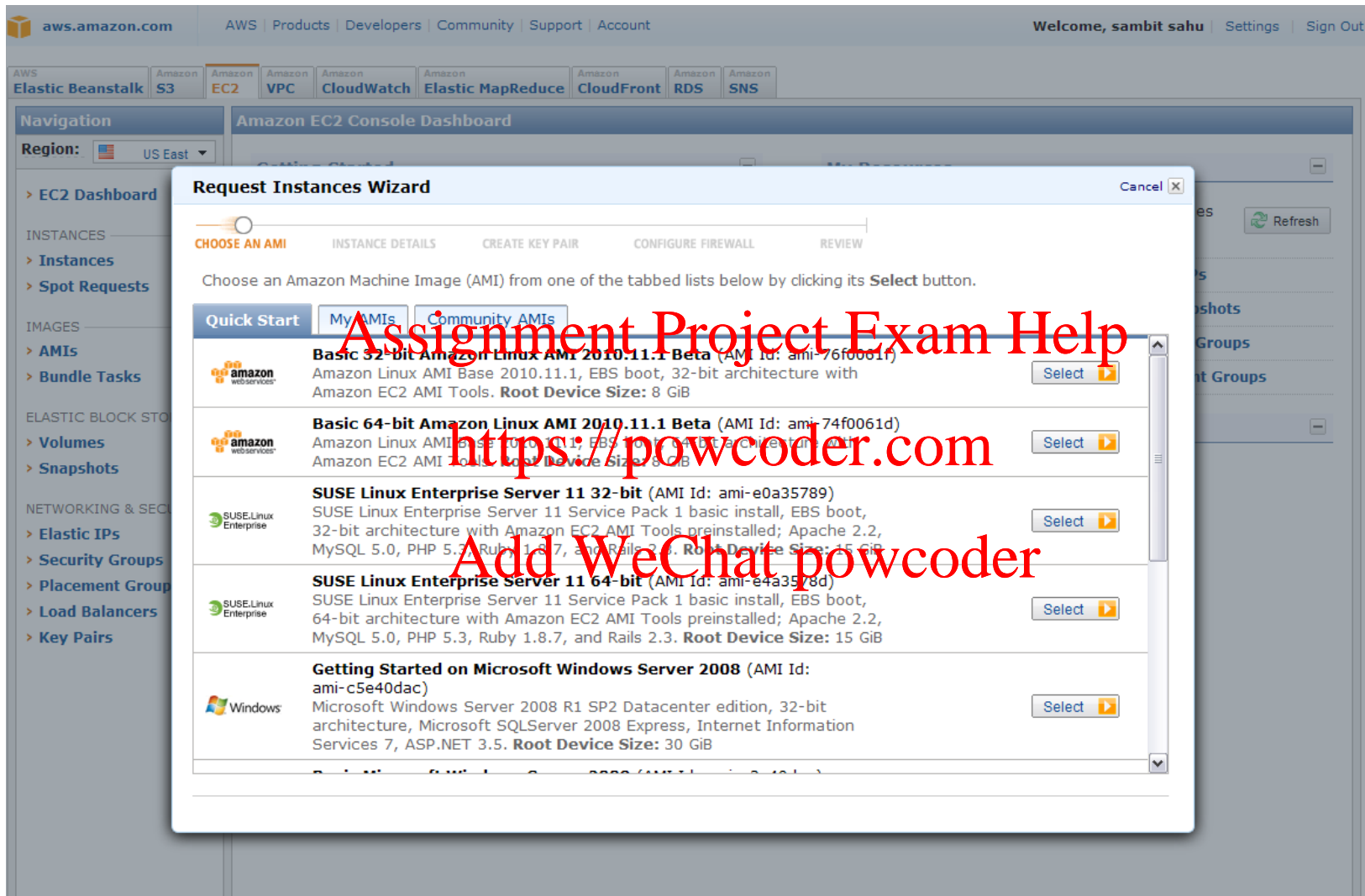
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- User logs in with AWS credentials

User launches request instance → a list of prebuilt stack is provided



- AWS shows a list of available pre-built base software stack (called **Virtual Appliances**) user may request to add to the machine

User can choose the resource size (CPU, mem choices)

Request Instances Wizard

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Availability Zone:** No Preference

Instance Type:

Type	CPU Units	CPU Cores	Memory
Micro (t1.micro)	Up to 2 ECUs	1 Core	613 MB
Small (m1.small)	1 ECU	1 Core	1.7 GB
High-CPU Medium (m1.medium)	5 ECUs	2 Cores	1.7 GB

Launch Instances EC2 Instances let you launch commonly large fixed capacity instances.

Request Spot Instance

Launch Instances Into Your Virtual Private Cloud

< Back Continue ▶

- Instance request wizard guides through resource choices

User specifies security/access configurations

The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, the user's name 'Welcome, sambit sahu', and links for 'Settings' and 'Sign Out'. Below this, a horizontal menu lists various AWS services: Elastic Beanstalk, S3, EC2, VPC, CloudWatch, Elastic MapReduce, CloudFront, CloudFormation, RDS, SNS, and IAM. The left-hand navigation pane is titled 'Navigation' and lists categories like 'Region' (set to US East (Virginia)), 'EC2 Dashboard', 'INSTANCES' (with sub-links for Instances, Spot Requests, and Reserved Instances), 'IMAGES' (with sub-links for AMIs and Bundle Tasks), 'ELASTIC BLOCK STORE' (with sub-links for Volumes and Snapshots), and 'NETWORKING & SECURITY' (with sub-links for Security Groups, Elastic IPs, Placement Groups, Load Balancers, and Key Pairs). The main content area is the 'Amazon EC2 Console Dashboard', which has tabs for 'Getting Started' and 'My Resources'. A modal window titled 'Request Instances Wizard' is open in the foreground. It has a progress bar with five steps: 'CHOOSE AN AMI', 'INSTANCE DETAILS', 'CREATE KEY PAIR' (which is the current step), 'CONFIGURE FIREWALL', and 'REVIEW'. The 'CREATE KEY PAIR' step contains a text box explaining that public/private key pairs allow secure connection to the instance and instructs the user to click 'Create & Download your Key Pair'. Below this, there are three radio button options: 'Choose from your existing Key Pairs' (which is selected), 'Create a new Key Pair', and 'Proceed without a Key Pair'. The 'Choose from your existing Key Pairs' option has a dropdown menu showing 'IM2011'. At the bottom of the wizard, there are '< Back' and 'Continue >' buttons. Overlaid on the screenshot in large red text are the phrases 'Assignment Project Exam Help', 'https://powcoder.com', and 'Add WeChat powcoder'.

AWS provisions an instance and returns user credentials

AWS Elastic Beanstalk S3 EC2 VPC CloudWatch Elastic MapReduce CloudFront CloudFormation RDS SNS IAM

Navigation

Region: US East (Virginia)

EC2 Dashboard

INSTANCES

- > Instances
- > Spot Requests
- > Reserved Instances

IMAGES

- > AMIs
- > Bundle Tasks

ELASTIC BLOCK STORE

- > Volumes
- > Snapshots

NETWORKING & SECURITY

- > Security Groups
- > Elastic IPs
- > Placement Groups
- > Load Balancers
- > Key Pairs

My Instances

Launch Instance Instance Actions Show/Hide Refresh Help

Viewing: All Instances All Instance Types 1 to 3 of 3 Instances

	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	Monitoring	Virtualization	Placement
<input type="checkbox"/>	empty	i-a1c318cf	ami-e4a3578d	ebs	t1.micro	running	IM2001	IM2011	basic	paravirtual	
<input type="checkbox"/>	MyFirstInstance	i-3b7aa155	ami-76f0061f	ebs	m1.small	terminated	default		basic	paravirtual	
<input checked="" type="checkbox"/>		i-6176ad0f	ami-e4a3578d	ebs	t1.micro	running	IM2001	IM2011	detailed	paravirtual	

Assignment Project Exam Help

1 EC2 Instance selected

EC2 Instance: i-6176ad0f

Description Monitoring Tags

AMI: sles-11-sp1-v1.00.x86_64 (ami-e4a3578d) **Zone:** us-east-1c

Security Groups: IM2001 **Type:** t1.micro

Status: running **Owner:** 026317314573

VPC ID: - **Subnet ID:** -

Source/Dest. Check: **Virtualization:** paravirtual

Placement Group: **Reservation:** r-ddfe6db1

RAM Disk ID: - **Platform:** -

Key Pair Name: IM2011 **Kernel ID:** aki-427d952b

Monitoring: detailed **AMI Launch Index:** 0

Elastic IP: - **Root Device:** sda1

Root Device Type: ebs **Tenancy:** default

Lifecycle: normal

Block Devices: sda1

Public DNS: ec2-50-16-69-93.compute-1.amazonaws.com

Private DNS: ip-10-196-229-93.ec2.internal

Private IP Address: 10.196.229.93

Launch Time: 2011-05-26 10:45 EDT

State Transition Reason:

Next Week

- Account setup and testing

- Sign up for AWS account. Sign up for AWS EC2 and S3 services.
- Create a micro instance with Amazon Linux stack with appropriate keys and access control using AWS portal. SSH into the instance you created.
- Read Chapter 1 and 3 from AWS in Action book.

Assignment Project Exam Help

<https://powcoder.com>

- Assignment 0

- Building Modern Web Application (Just complete Module 10) by following this link:
<https://aws.amazon.com/getting-started/hands-on/build-modern-app-fargate-lambda-dynamodb-python/>

Add WeChat powcoder

Some additional links

- [Hands-on Tutorials on AWS: https://aws.amazon.com/getting-started/hands-on/](https://aws.amazon.com/getting-started/hands-on/)
- <https://aws.amazon.com/solutions/case-studies/>
- <http://aws.amazon.com/awscredits>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder